

(51) Int.Cl.⁹
G 0 6 F 17/60

識別記号

庁内整理番号

F I

9168-5L

G 0 6 F 15/21

Q

審査請求 未請求 予備審査請求 有 (全 55 頁)

(21) 出願番号 特願平6-520253
 (86) (22) 出願日 平成6年(1994)3月4日
 (85) 翻訳文提出日 平成7年(1995)9月11日
 (86) 国際出願番号 PCT/US94/02468
 (87) 国際公開番号 WO94/20912
 (87) 国際公開日 平成6年(1994)9月15日
 (31) 優先権主張番号 08/028,360
 (32) 優先日 1993年3月9日
 (33) 優先権主張国 米国 (US)

(71) 出願人 キャッツ ソフトウェア インコーポレイ
 テッド
 アメリカ合衆国 94303 カリフォルニア
 州 パロ アルト エンバーカデロ ロー
 ド 1731
 (72) 発明者 クレックナー, ジェイムス, イー.
 アメリカ合衆国 94301 カリフォルニア
 州 パロ アルト カウパー ストリート
 1855
 (74) 代理人 弁理士 谷 義一 (外1名)

最終頁に続く

(54) 【発明の名称】 金融手段を作成、構成、操作および評価するオブジェクト指向システム

(57) 【要約】

金融手段を構築し、管理するシステムであり、金融アプリケーションを支援する関数及びデータ型のライブラリからなる金融フレームワークを含んでいる。システムにおいては、ユーザが、任意の複雑な手段を曖昧性がなく、定義し、価格付けし、評価することができ、金融手段を他の手段とともにポートフォリオとして管理して、利益と損失、資本要求 (capital requirement)、コスト、信用開示 (credit exposure) を評価し、報告書の要求を取り扱い、リスクを管理し、規則化された要求を合致させ、手段及び取引に関係する全ての文書を安全な環境で作成することを支援することができる。金融手段は、相互に関連付けられたチームのネットワークにより構成され、他の金融手段の一部となることができる。

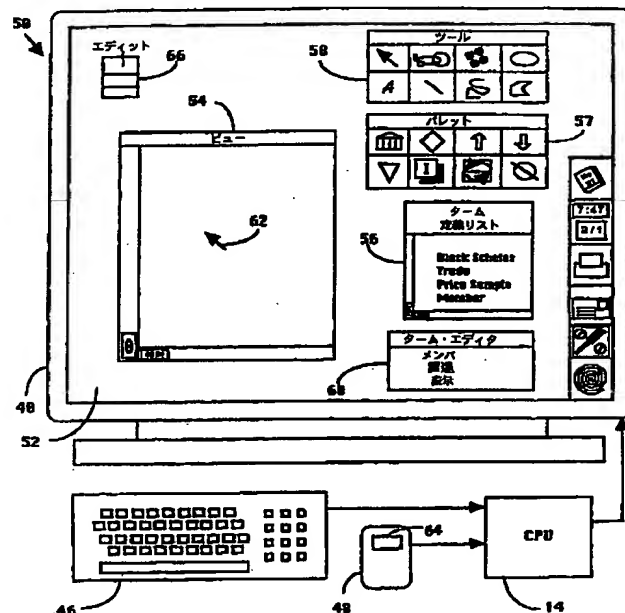


Fig. 5

【特許請求の範囲】

1. 金融ターム定義を構築し、前記金融ターム定義を金融タームとしてインスタンス生成し、前記金融タームを評価するためのコンピュータで実現するシステムにおいて、

コンパイルされたターム定義とゼロ以上の前に構成した金融ターム定義を有し、ターム定義 (506) の空でないライブラリ (504) を前記コンピュータのメモリに格納し、

前記ライブラリから金融ターム定義 (56) を選ぶ手段と、

前記ライブラリからターム定義を選択し、タームとしてインスタンス生成する手段で、前記選ばれた金融ターム定義 (57) 内に前記選択されたターム定義があり、

複数のそのような前記タームの間の交信を確立し、それにより、選ばれた金融ターム定義 (60) を構成する手段と、

前記ライブラリ内に前記選ばれた金融ターム定義を格納する手段 (66) と、

前記選ばれた金融ターム定義を金融タームとしてインスタンス生成する手段 (57) と、

前記金融タームを評価して、金融タームに関する少なくとも1つの選択された値を生じる手段 (712) と
を備えるシステム。

2. 請求項1記載のシステムにおいて、複数のそのようなタームの間を、それらの間の確立された交信により、新しい金融ターム定義 (54) として表示する手段と、前記新しい金融ターム定義をそのような前記タームのネットワーク内に、それらの間の交信によりインスタンス生成する手段をさらに備えるシステム。

3. 請求項1記載のシステムにおいて、前記タームは型 (710) を有し、適合する型のターム間で交信を確立することができ、非適合である型のターム間では交信を確立することができなくする手段 (708) をさらに備えるシステム。

4. 請求項1記載のシステムにおいて、各前記タームは型を有し (710) 、第1と第2の他の前記タームがそれへの確立した交信を有することできる少なくとも

1つのタームを備え（706）、さらに、前記第1及び第2の他のタームが適合する場合、それらのターム間で交信を確立することができ、前記第1及び第2の他のタームが非適合である場合、前記第1及び第2の他のターム間では交信を確立することができなくする手段（708）を備えるシステム。

5. 請求項1記載の金融ターム定義を構成するシステムにおいて、ターム定義（50）を選択し、前記ターム定義からタームとしてインスタンス生成し（57）、前記ターム間のグラフィカル関連を確立し、これらの間の交信を表す（60）グラフィカル・ユーザ・インターフェースの部分をもさらに備えるシステム。

6. 請求項5記載のシステムにおいて、少なくとも1つのウィンドウ領域を有し、そこに、ターム定義がインスタンス生成される（54）ディスプレイ装置（50）を備えるシステム。

7. 請求項1記載のシステムにおいて、前記金融タームは金融手段を表し、さらに、前記金融タームを評価する前記手段は、前記金融手段の少なくとも現在の正味の価値（net present value）を生成することができるシステム。

8. 請求項1記載のシステムにおいて、少なくとも1つの前記タームが、他の前記タームに処理のために、前記確立された交信を介して渡すことのできるプライベート・データを含み、前記プライベート・データをアクセス権限を有しないユーザにより修正することができる手段をもさらに備えるシステム。

9. 請求項1記載のシステムにおいて、前記選ばれたターム定義を修正することをさらに備えるシステム。

10. 請求項1記載のシステムにおいて、複数のそのような前記ターム間の前記交

信を確立する手段は、前記金融タームを相互に交信するコンピュータ・プロセスとして非同期的に動作させる手段を表しているシステム。

11. 金融ターム定義を構築し、前記金融ターム定義を金融タームとしてインスタンス生成し、前記金融タームを評価するためのコンピュータで実現するシステムにおいて、前記システムは、コンパイルされたターム定義とゼロ以上の前に構成した金融ターム定義を有し、ターム定義の空でないライブラリは前記コンピュータのメモリに格納され、

前記ライブラリから金融ターム定義を選ぶ (56) ステップと、

前記ライブラリからターム定義を選択し、タームとしてインスタンス生成するステップで、前記選ばれた金融ターム定義 (57) 内に前記選択されたターム定義があり、

複数のそのような前記タームの間の交信を確立し、それにより、選ばれた金融ターム定義 (60) を構成するステップと、

前記ライブラリ内に前記選ばれた金融ターム定義を格納するステップ (66) と

、
前記選ばれた金融ターム定義を金融タームとしてインスタンス生成するステップ (57) と、

前記金融タームを評価して、金融タームに関する少なくとも1つの選択された値を生じるステップ (712) と

を備える方法。

【発明の詳細な説明】

金融手段を作成、構成、操作および評価するオブジェクト指向システム

背景技術

本発明は、金融 (finance) の分野に関し、この分野内において、特にコンピュータに実装された金融手段 (financial instrument) および取引 (transaction) を構成、管理するシステムに関する。

「金融手段 (financial instrument)」 (又は手段: instrument) は、ここにおいて、金融取引の基礎となっている構造、例えば商品またはサービスの売買、ボンド (bond)、エクイティ (equity)、外国為替契約又は商品 (foreign exchange contract or commodity)、通貨 (currency)、インデックス (index) 又は、スワップ (swap) やオプション (option) のような他の手段のデリバティブ (derivative) 等のことを意味している。デリバティブ手段の詳細な議論については、Mark, Susan Ross DERIVATIVE FINANCIAL PRODUCTS (HarperBusiness 1991) を参照されたい。この文献は、参照により、この明細書に組み込まれている。

特定の手段、例えばデリバティブは、使用が急速に増加している。例えば、1980年代では、銀行の資本は、あったとしてもあまり、デリバティブとは結びついていなかった。今日では、銀行の資本の半分以上が、複雑に構成されている投資および他の取引と結びついている。

一般的に、金融手段は、構成された取引を表している。この構成された取引には、外国通貨交換、レート・インデックス、コンティンジェント・イベント等があり、他の手段の一つ又はポートフォリオのパフォーマンスに依存している。金融手段のパワフルな特徴は、それが表している金融取引の「管理」の面であると思われる。この管理の例は、価格付け、パフォーマンスの追跡、ヘッジング、およびクーポン支払いのような関連する債務の監視等がある。

これらの金融手段の管理は、現在主として、キャッシュ・フロー又は、パラメータ・テクニックにより操作されており、キャッシュ・フローの集合又は1組

のパラメータとして手段を表現している。たとえば、スプレッドシートは、分析したり、パラメータのパフォーマンスを追跡したりするのに、よく用いられてい

る。他の手段のパフォーマンスを基礎としている手段を分析するのに、基礎となっている手段をまずスプレッドシートの方法で分析し、それから、デリバティブ手段に対して要求されているように評価を一つにまとめる。たとえば、手段のポートフォリオのパフォーマンスを評価するために、構成する手段に対する個々のスプレッドシートを統合して一つにまとめる。

スプレッドシートの使用は、基礎としている手段が合理的な数の場合にのみ実用的である。たとえば、ある銀行機関は、スプレッドシートは同時には5つの取引までにのみ使用できるという規則を有している。この数以上の取引に対しては、カスタム・ソフトウェア・システムを使用しなければならない。しかしながら、ある機関は、任意の数の基礎となる手段を有することができる。そして、さらに、これらの基礎となる手段がそれ自身他の複数の手段から構成されているかもしれない。このように、数百、数千の基礎となるスプレッドシートを有するかもしれない、ポートフォリオを管理するために一つにまとめる必要があるポートフォリオの日々の処理が、本質的に排除されている。

スプレッドシートの一般的な解決方法に加えて、専門的な製品やシステムが様々な機関およびその管理部門に対して開発されている。この専門的な製品の例として、SwapWare（商標）（利率と通貨スワップ契約、借り入れ、および外国為替契約を組織化し及びモデル化する）、CapWare（商標）（キャップ、フロアー、カラー、先物相場を価格付けし及びモデル化する）、Equity Derivatives（商標）（エクイティ・スワップ取引を組織化し及びモデル化する）、Strike（商標）（スワプション（swaption）及びカウンタ・ボンド・オプションを組織化し及びモデル化する）などがあり、これらは、カリフォルニア州パロアルトにあるC*ATS Software社から販売されている。ポートフォリオの要素、たとえばポートフォリオに関連する将来の契約は、Future（商標）により管理できる。そして、再評価、利子付け、キャッシュ管理、継続期間、凸面、ギャップ分析、ヘッジング、感度及び他の分析は、様々な特定の製品を横断してC*ATS社からのRisk Manager（商標）を用いて行うことができる。

これらの専門的な製品やシステムは、特定の手段の特定の網目に対して処理を

行うことができるが、これらは、相当程度再プログラミング及びコンパイルをせずには、一般的に新しい別の手段に対して取り扱うように拡張することができない。むしろ、新しい手段を構成し（すなわち、そのターム、パフォーマンスの特徴等を定義かつ例示し）、新しい専門化したシステムを開発する必要がある、これは、高価で、時間がかかる過程である。新しくかつ異なる手段に適応させるためにシステムを拡張する能力は、ここで詳しく説明されている。

その上、ある従来のシステムは、選択された手段の評価を提供しており、他のシステムは手段のパフォーマンスの追跡を適用している。そして、さらに他のシステムは、これらの手段に関連する債務を監視することを提供している。一方、ある従来のシステムは、複数の能力を提供している。現在入手可能なシステムのどれ一つとして、統合した様式で、これらの各面を提供しておらず、人間とコンピュータの両方に対して通信コストを減少していない。これらの様々な面を一緒に結びつける能力は、ここでは緊密な統合と呼んでいる。

この分野の他の制限は、従来のシステムが新しく異なる手段を取り扱うように修正したり、新しいシステムを作成するためには、ソフトウェア開発に対する比較的高いレベルの専門性が要求される。これは、専門化された製品及びサービスは、投資機関に対して外部の組織又は投資機関自体のソフトウェア開発グループにより開発されることを意味する。かつまた、従来のシステムが新しい型の手段を取り扱うように修正することは、第三者や内部のプログラマの領域であった。投資家は、大部分、コンピュータ・プログラミングの実質的な訓練と経験が欠落しているために、新しい手段を管理したり組織化する目的で、新しいシステムを作成したり、従来のシステムを修正したりすることから閉め出されている。投資家の新しい手段を組織化したり管理したりする能力は、その投資家を市場において他の投資家と差別化するので、開発の速度と使用の容易性は重要である。

結局、金融の分野におけるこれらのものは、金融手段を組織化し、管理するための最適なツールを提供していない。

発明の概要

本発明は、金融手段を組織化しかつ管理するためのシステムである。この発明

は、金融アプリケーションをサポートする関数のライブラリ及びデータ型で構成されている金融フレームワークを含んでいる。処理能力を構築できる構成要素を提供している。システムは拡張可能あるので、新しい任意の複雑な型の手段に適用することができる。そして、緊密に統合されているので、統合した方法で手段に対して様々な仕事を実行することができる。システムは、ユーザが、曖昧ではなく定義し、価格付けし、任意の複雑な手段を再評価し、手段をポートフォリオ中の他の手段と共に、利益と損失と資本要求 (capital requirement) とコストと信用開示 (credit exposure) とを経理して評価し、かつ、要求を報告し、リスクを管理し、規則の要求を満たし、そして手段と手段に含まれる取引に関連するすべての文書を作成するために、安全な環境で管理することができる。手段を組織化しかつ管理するシステムは、一つの一貫したフレームワークから拡張可能であるので、新しい手段は、確立した手続きで定義することができ、新しい手続きは従来の手段と統合することができる。

金融フレームワークは、全てのソフトウェア・ツールを特定の計算環境から隔離し、かつグローバルに全てのツールに使用する基本的機能を統合している。これは、本発明はコンピュータ・プラットフォーム独立であることを意味している。

金融フレームワークにより提供される3つの重要なサービスは、データベース管理、更新管理、そしてターム評価エンジンである。データベース管理サービスには、データ・モデル、アプリケーション・プログラミング・インターフェース (API)、及びデータ持続性 (data persistence 通常はデータのセーブとリストアと呼ばれる) が含まれる。データ・モデルは階層的に、ライブラリ、定義、ビュー、及びオブジェクトの4レベルに編成されている。これらのレベルとを介して、データ・モデルは、システム中の各オブジェクトをユニークに定義することができ、オブジェクトは必要に応じて再使用することができる。APIは、各オブジェクトに対応するプログラミングのインターフェース及びデータ・モデル中

の他のオブジェクトとの関係を提供し、新しいタームをダイナミックにランタイム時にシステムに追加することができる。APIは、さらに外部アプリケーション

を本発明の金融フレームワークと統合することができ、従来のアプリケーションとシステムは、本発明と協力して働くことができる。最後に、APIは、どんなデータ型の定義や構造でも検索する能力をランタイム時に提供することができる。データ持続性は、データベース独立の方式でデータを格納したり、検索したりする能力を提供する。

更新管理サービスは、ターム間の「リンク」を提供し、システム中の全てのデータがアップデートできるようにしている。値が他のオブジェクトの値に従属しているオブジェクトは他のオブジェクトとリンクしており、他のオブジェクトの一つが修正されたとき、リンクしているオブジェクトは変化を知らされ、その値は適切に修正され得る（すなわち、ダイナミックに、またはユーザの要求のより）。

ターム評価エンジンは、相互に関連している（interconnected）タームのネットワークを評価するメカニズムである。全ての要求されたターム入力、タームの出力が有効になる前に（すなわち、この出力が計算される前に）、評価される必要がある。評価は、入力の順序と独立であり、これは、評価は分割されてもよいことを意味している。評価の各部分は、独立に行ってもよく、たとえば、複数のプロセッサ（複数のプロセス）を用いて、金融手段で以前に可能であったより、相当大きな量のデータと分析を取り扱うことができる。

本発明によれば、手段は、複数の相互に関連したタームから構成されている。タームは、少なくとも記憶位置、記憶位置に値を入れる手段、及び出力関数を含み、そして、他の関数、1以上の入力、プライベート・データ等を含むことができる個別のオブジェクトである。ある実施例では、タームは、カプセル化したメソッド（すなわち、オブジェクト指向プログラミング言語で実現された実施例において）である。タームはそれ自身他のタームで作られている。タームはネットワーク状に一緒に関連しており、一つのタームからの適切な出力値は、他の用語に入力値として渡される。

手段を構成しているタームは、一緒になって、システムが少なくとも単一の通

貨に対する手段の現在の正価（Net Present Value:NVP）を定めることができる

ために十分なデータを提供する。また、システムが、任意の定義された通貨に対するNPV、部分デリバティブ、たとえばデルタ（特定の重要な変数（specified significant variable）の価格に関する第1のデリバティブ）、ガンマ（特定の重要な変数の価格に関する第2のデリバティブ）、シータ（時間に関する価格の第1のデリバティブ）、及びヴェガ（ボラタリティ（volatility）に関する第1のデリバティブ）、支払いのリスト、利益と損失情報、市場会計価格（mark-to-market accounting value）等を作成することができるデータをさらに提供することができる。

各タームに対して、そのタームに関連したタームから入力値を受信し、システムは、自動的に入力値の型をチェックし、その関連が許可できるかを定める。要求される入力値の型が供給された出力値の型とが不適合である2つのタームを関連させる試みが行われた場合は、システムはユーザに警告する。たとえば、日付のデータ型を提供するタームが、たとえば浮動小数点数のデータ型を入力として期待している平方根の数学的なタームに関連付けられた場合、システムはユーザに、型が不適合であるため、このような関連付けは不適合であると警告する。この種の型チェックは、ここでは強い型チェックと呼ばれる。

さらに、ある種のタームは、色々な型の演算を行うことができ、このため、色々な型の入力値を受け入れることができる。たとえば、“if-then-else”のタームは、“if”条件は任意のデータ型を出力として提供するとテストしてもよい。これらのタームに関して、“if”条件入力、Booleanすなわち、“true”又は“false”として評価する必要がある。しかしながら、if-then-elseの用語の出力は、“then”及び“else”の入力により特定された任意の型でよい。両方の入力は、同じ又は適合できる型である必要がある。システムはユーザに、要求される値の型と不適合であるタームに入力値を関連付ける試みが行われたかを警告する。上述するように、タームから入力又は／及び出力の値の型を確立することは、ここでは、推論された型チェック（inferred type checking）と呼ばれる。

様々なメソッドは、本発明に従って手段を組織化するために使用される。たとえば、手段は、グラフィカル・ユーザ・インターフェース・システムから供給さ

れるグラフィカルな入力ツールを使用し、特別な手段定義言語を使用し、スプレッドシート・システムを使用し、パラメータ入力フォームを使用する等で構成される。

グラフィカル・ユーザ・インターフェースを利用するある実施例では、ディスプレイ・ウィンドウが供給され、手段のグラフィカル表示がビュー・ウィンドウ内に構築される。ターム定義のリスト（ここでは「ターム・リスト」と呼ぶ）を含む他のウィンドウは、ユーザが選択する複数のそのようなターム定義を提供する。各ターム定義に関連して、アイコンと、たとえばテキスト・フィールド等のグラフィック・オブジェクトから構成されているターム・グラフィックがある。ターム・エディタは、メンバー・エディタ、関連エディタ、ディスプレイ・エディタ等で構成することができ、タームの外観を構成することに使用され、ターム間の関連付けを行い、タームの値をエディットするのに使用する。このように行うとき、ソフトウェアは、ターム間で値を渡すためのような実際のターム間の関連を作成する。

図面の簡単な説明

本発明は、本発明の実施例を含めて、図面と関連して詳細に以下に記述されている。

図1は、本発明が動作する汎用コンピュータ・アーキテクチャを示す図である。

図2は、本発明の実施例に従ったシステムのアーキテクチャのある面を示すエンティティの関連図である。

図3は、本発明の実施例に従って、定義を作成するのに使用する複数のユーザ・インターフェース・アプリケーションと、金融フレームワークとの関係を示す図である。

図4は、本発明の実施例に従って、ターム、メンバー、及び入力、出力のためのデータ構造かつデータ格納を示す図である。

図5は、本発明の実施例の視覚的特徴とグラフィカル・ユーザ・インターフェースのレイアウトを示す図である。

図6は、本発明の実施例の一例を構成する様々な手段の間の関係を示す図である。

図7は、本発明の実施例の一例の部分形成するエクイティの過去、現在及び未来の価格例のテーブルを示す図である。

図8は、本発明の実施例の一例のエクイティ手段を構成するためにあるステップを行った後のユーザ・インターフェースのある部分の状態を示す図である。

図9は、図8に示したエクイティ手段のターム・グラフィック表示を示す図である。

図10は、本発明の実施例の一例のエクイティ・スプレッド手段の構成中で、あるステップを行った後のグラフィカル・ユーザ・インターフェースのある部分の状態を示す図である。

図11、本発明の実施例の一例のエクイティ・スプレッド NPVターム定義の構成中で、あるステップを行った後のグラフィカル・ユーザ・インターフェースのある部分の状態を示す図である。

図12は、図11に示したエクイティ・スプレッド NPVターム定義のターム・グラフィック表示を示す図である。

図13は、本発明の実施例の一例のエクイティ・スプレッド・ボラタリティ・ターム定義の構成中で、あるステップを行った後のグラフィカル・ユーザ・インターフェースのある部分の状態を示す図である。

図14は、図13に示したエクイティ・スプレッド・ボラタリティ・ターム定義のターム・グラフィック表示を示す図である。

図15は、図10に示したエクイティ・スプレッド手段のターム・グラフィック表示を示す図である。

図16は、本発明の実施例の一例のエクイティ・スプレッド・コール・オプション手段の構成中で、あるステップを行った後のグラフィカル・ユーザ・インターフェースのある部分の状態を示す図である。

図17は、図16に示したエクイティ・スプレッド・コール・オプション手段のターム・グラフィック表示を示す図である。

図18は、本発明の実施例の一例のAjax-Monolithicエクイティ・スプレッド・コール・オプション手段の構成中で、あるステップを行った後のグラフィカル・ユーザ・インターフェースのある部分の状態を示す図である。

図19は、本発明の実施例に従って、外部タームにアクセスするメソッドを示す図である。

これらの図面の間で、同様の参照番号は同様の要素を示している。

詳細な説明

説明のため、本発明は、選択された例（アプリケーション）で、本発明の複数適切な実施例を参照して記載される。この技術の当業者にとっては、例とそれらが開発された環境は、本発明の機能と特徴を示していることが自明であろう。この分野の当業者にとっては、例に対して、変形、追加、削除、修正等を考えることは自明であろう。また、これらの例を示していることは、本発明の範囲を限定するものではない。

図1には、本発明が動作する種類のシステムの汎用のアーキテクチャ10が示されている。アーキテクチャ10は、メイン・バス12を備えており、それは、色々な構成要素を相互接続している。その構成要素は、少なくとも以下のものを含んでいる。中央演算処理ユニット（CPU）14、浮動小数点演算ユニット（FPU）16、集積回路プロセッサ（ICP）18、バス・コントローラ20、ビデオRAM22、ダイナミックRAM（DRAM）24、スタティックRAM（SRAM）26、デジタル信号プロセッサ（DSP）28、内部ハード・ディスク30、外部メモリ装置32（たとえば、SCSIポート38を介して接続されている）、外部ネットワーク装置36たとえば、イーサネット・ネットワーク38で、SCSIポート34を介して接続されている）、（CRTのような）ディスプレイ装置40、プリンタ42（ポストスクリプト装置のようなもので、たとえば、シリアル・ポート44を介して接続されている）、キーボード46、及び（トラックボール、マウスのような）ポインティング機器48ものがある。上記の各要素は、

この技術の当業者にとってよく知られており、特定の構成要素の機能と重要性は適切なときにさらに詳しく記載される。

多くの現在のコンピュータ・システムは、グラフィカル・ユーザ・インターフェース (GUI) を支援している全点アドレス可能 (all-points-addressable APA) 又は、ビットマップ表示スクリーンを採用している。よくあるGUIのプラットフォームは、アップル・マキントッシュ・シリーズのコンピュータ、マイクロソフト・ウィンドウを実行しているパーソナル・コンピュータ、NeXT社が製造しているNeXTシステムを含む。これらのシステムでは、ユーザは、キーボード及びマウスのようなポインタ制御メカニズムを使用して、コンピュータと対話している。一般的には、ユーザのソフトウェア・アプリケーションとの対話の結果は、ウィンドウと呼ばれる画面の領域内に表示する。ソフトウェア・アプリケーション・パッケージ、文書、ファイルのようなものは、アイコンとして表示され、ポインタをアイコン上に並べて、マウス上のボタンをダブル・クリックすることで、立ち上げる、すなわち開く。

NeXTシステムに対するユーザ・インターフェースに焦点をあてると、これは、2つの要素であるNeXTStepとディスプレイ・ポストスクリプトで構成されている。NeXTStep要素それ自身は、ワークスペース・マネージャ、インターフェース・ビルダ、アプリケーション・キット、ウィンドウ・サーバの4つの構成要素から成り立っている。ワークスペース・マネージャは、ファイルやディレクトリを表示するユーザ・インターフェース構成要素で、アイコン・ドック (アプリケーション・ソフトウェアを表すアイコンが位置している表示画面の領域) 及びアプリケーションの立ち上げを管理している。インターフェース・ビルダは、アプリケーション設計者がアプリケーションのユーザ・インターフェースをレイアウトすることができるユーザ・インターフェース構成要素である。アプリケーション・キットは、ウィンドウ及びボタンのようなグラフィカル・オブジェクトのソフトウェア・ライブラリである。ウィンドウ・サーバは、表示画面を管理し、様々な実行中のプログラムに対する描画要求を取り扱い、マウスのクリックやキー押下のようなイベントをどのプログラムに対して知らせるかを定めるユーザ・インターフェース構成要素である。ウィンドウ・サーバは、次に、ディ

スプレイ・ポストスクリプトを呼び出し、実際の表示装置やプリンタに対する描

画要求を実行する。(NeXTStepと関連する要素のもっと詳細な記述は、Webster “The NeXT Book” (Addison-Wesley 1989) を参照 この本は参照によりここに組み入れられる。)

図2を参照すると、ここには、エンティティ関係図500が図示され、高いレベルの観点から、本発明の実施例の1つのシステムにおけるデータ構造を示している。エンティティ関係図は、オブジェクト指向ソフトウェア・システムと共通的に関連している。しかし、このような図に表されているアーキテクチャは、他のシステムに適用可能である。図2に示されているボックスは、エンティティに対応している。

エンティティ関係図は、ソフトウェア・システムの高いレベルのデータ構造の詳細について示している。ひとつの実装方法は、C++プログラミング言語のクラスを使用することで、エンティティ及びポインタ又は関係を表すポインタのようなデータ型を表す。この技術の当業者はよく理解できるように、プログラマが定義したクラスは、オブジェクトの詳細を隠蔽することができ、クラスにより定義された操作の集合を用いてデータにアクセスするデータ抽象を提供する。(Ellis, Margaret and Bjarne Stroustrup “The Annotated C++ Reference Manual” (Addison-Wesley 1990) 及びCoplien, James O., “Advanced C++ Programming Style and Idioms” (Addison-Wesley 1992) を参照。これらは参照により、ここに組み込まれる。) エンティティ構造の例は、図2に示されており、全てのエンティティは、エンティティのエンティティ502から導出され、そのプロパティを継承している。エンティティのエンティティ502は、たとえばオブジェクトが作成するメソッド、ダイナミック・メモリ・アロケーション及びデアロケーション、格納と検索のメソッド等を提供する。

本発明の実施例に従ったユーザ・インターフェースはイベント駆動としてもよいことを理解すべきである。イベントは、処理を要求する、重要で非同期のシステム中の変化である。イベントの例は、マウス(図示せず)のようなポインティング機器上のボタンのクリックである。システムはイベントによりインターフェースしているので、ユーザ・インターフェースはイベントにより駆動され

る。下記の評価エンジンは、しかしながら、要求駆動である。すなわち、イベントがシステムの動作を開始し、システムによる重要な処理活動の後、システムが値を返すまで、これ以上のイベントはない。評価の間、システムのタームは他のタームからの値を、自分自身の処理に必要なとき「要求」する。更新通知は、イベント駆動で、値への更新の通知は、その値が採用される各タームへ伝達される。適切なコンテキスト中で、評価それ自身はイベント駆動であることができ、ここで、エンティティのエンティティ502は、たとえば、イベントを取り扱うメソッドを提供してもよいことを理解すべきである。さらに、インターフェースはイベント駆動である必要はない。

図2において、「=」の記号は、“is a”関係を表している。“is a”関係は、あるクラスのオブジェクトの定義が他のクラスのオブジェクトの定義のバリエーションであることを意味している。（“is a”関係は、C++のような言語の継承の概念と同様のものである。しかし、オブジェクトの性質の詳細とクラス間の関係をあまり含まれていない。）「=」に最も近いエンティティは、ベース・エンティティに対応しており、これから他のエンティティが情報を継承する。「=」から最も遠いエンティティ（又は複数のエンティティ）は、導出されたエンティティである。たとえば、ライブラリ・エンティティ504はエンティティ502と“is a”の関係にあるエンティティである。すなわち、ライブラリ・クラスの全てのメンバーはエンティティ・クラスのメンバーである。同じことは、定義エンティティ506、ビュー・エンティティ508、及びオブジェクト・エンティティ510にもいえる。これらのクラスの各メンバーは、エンティティ・クラスのメンバーでもある。

図2の単一のやじりを有する矢印は、単一の関係を示している。たとえば、エンティティのエンティティ502から定義エンティティへの単一の矢印は、ただ一つのエンティティのエンティティ502の定義しかないことを意味している。（すなわち、各エンティティは、ユニークに定義されている。）2つのやじりがある矢印は、複数の関係を示している。たとえば、ライブラリ・エンティティ504と定義エンティティ506との間を指している2つのやじりがある矢印は、ライブラリ・エンティティが複数の定義を有してもよいことを意味している。

単一の又は2つのやじりのある矢印は、それに関連してオプションの「O」または「|」を有してもよい。「O」（オプション的）は、ゼロのエンティティ（空である）を許している関係であることを意味している。「|」は、関係において、正確に一つのエンティティを有することを意味している。たとえば、ライブラリ・エンティティ504を定義エンティティ506の間の単一のやじりを有する矢印の側に、それに関連して「|」を有しているが、これは、つねに少なくとも1つのライブラリ・エンティティ504が定義エンティティ506に対して存在していることを意味している（すなわち、定義はつねに1つのライブラリ中に存在している。）。

本発明によるシステムは、Smalltalkのようなランタイム型のシステムを実装している。（Goldberg, Adele” Smalltalk-80: The Interactive Programming Environment (Addison-Wesley 1985) Pinson, Lewis J. and Richard S. Wiener, ” Object-C Object-Oriented Programming Techniques (Addison-Wisley 1991) を参照）ランタイム型のシステムは、システム中のデータの型を記述する特別なオブジェクトを使用し、データに対して、作成、削除、コピー等のような共通の操作を行うことができる。この特別のオブジェクトは、時として、「メタクラス」オブジェクトと呼ばれる。このシステムでは、これは「定義」オブジェクトと呼ばれる。定義は、型に含まれるサブの型、その型に対する操作を行う関数（又はメソッド）、及び関数が要求する入力データのような型特有の情報をアクセスするのに使用される。

複数のインターフェース・アプリケーションを定義を作成するのに使用してもよい。図3は、グラフィカル・エディタ602、スプレッドシート・アプリケーション604、手段定義言語（Instrument Definition Language: IDL）インターフェース606、及びフォーム・エントリ・インターフェース608を含むこのようなアプリケーションのいくつかを図示している。金融フレームワークのフレームワーク・プログラミング・インターフェース（FPI）610を有するインターフェースは、適切なコンパイラ612, 614, 616, 618を介して完成される。金融フレームワークは、データベース620にインターフェースする。

アプリケーション・プログラミング・インターフェース（API）の図2のデー

タ・モデルを実装するC++クラスに対する仕様は、付属する付録Aに詳細に記載されている。また、付録Bとして付属するのは、本発明の実施例の一つに従って、メカニズムを実装するのに適したデータ構造の記述である。この各付録において、接頭語“Cats”は、ソフトウェア・パッケージを特定しており、ここにリストされているコード、命令又はデータを他の部分を形成しない。このように、接頭語はAPIと構造の制限を形成しない。

各定義は、「名前付けられた」ビュー・エンティティ (view) 508を含んでいる。オブジェクトの名前は、パス名と同様のもので、上記のデータ構造モデルに従っている。たとえば、Library_A:Definition:First:View_Interface:Object_Payment_Listは、“Library_A”と名付けられたライブラリ内の“Definition_First”と名付けられた定義内の“View_Interface”と名付けられたビュー内の“Object_Payment_List”と名付けられたオブジェクトを示している。エンティティの名前は、そのスコープ内でユニークに定められている。このように、2つのライブラリは同じ名前を持たず、同じライブラリ内の2つの定義は同じ名前を持たず、同じ定義内の2つのビューは同じ名前を持たず、同じビュー内の2つのオブジェクトは、同じ名前を持たない。各ビューは、たとえばインターフェース、モデル、アイコン又はエディタのような型を有している。ひとつの定義のインターフェース・ビューは、定義のアプリケーション・プログラミング・インターフェース (API) を宣言している。このため、本発明のこの実施例では、各定義は、インターフェース・ビューを有するように要求され、定義にはただ一つのインターフェース・ビューが許される。インターフェース・ビューを介して、定義の入力及び出力のメンバーはアクセスすることができる。

各ビューは、名前付けられたオブジェクト・エンティティ (オブジェクト) 510で構成されている。オブジェクト名は、それが含まれるビューのスコープ内でユニークである。一つのビュー内のオブジェクトには、3つの主要なサブクラスがある。ターム・エンティティ (ターム) 512、I/Oエンティティ (I/O) 514及びグラフィック・エンティティ (グラフィックス) 516である。手段を構成している主要な構成ブロックはタームである。タームは、タームの定義のメンバー・エンティティ (メンバー) 522により宣言される1以上のデータ・バッファを含

んでいる。(オブジェクトのベース・オブジェクト上の“is a”関係に従っている。)

ここで使用されているように、入力又は出力はタームとメンバーの組み合わせであり、タームの定義中に入力又は出力としてマークされている。このように、一つのメンバーは、入力として、出力として、又は入力と出力として使用される。入力と出力のメンバーは、定義のパブリック・インターフェースとして宣言される。ある定義がタームとしてインスタンス生成される場合は、その定義のパブリック・インターフェースは、タームの入力と出力となる。単一の入力又は／及び出力としてマークされた全てのメンバーは、現在の実施例中のアクセスの便宜のため、定義に対するインターフェース・ビュー内に含まれる。

図4に関し、一つのタームの入力を他のタームの出力と関連付ける例である。両方のタームは、ビューのLibrary:Example:Interface 702に存在している。”ToTerm”と名付けられたターム706は、”FromTerm”と名付けられたターム704と関連付けられた一つの入力710を有している。FromTerm 704は、Library:FromTerm:Interface 720のビューにより作成された定義ライブラリ:fromTerm (図示せず)のインスタンスである。Library:FromTerm:Interface 720は、”OutMember”と名付けられ、IsOutput値をTrueとすることで出力としたメンバー722を含んでいる。

同様に、ToTerm 706は、ビューのLibrary:ToTerm:Interface 730から作成されたLibrary:ToTerm (図示せず)の定義のインスタンスである。Library:ToTerm:Interface 730は、正確に1つの”InMember”と名付けられ、IsInput値をTrueとすることで入力としたメンバー732を含んでいる。

FromTerm 704とToTerm 706間の関連は、入力データ構造710を用いて行う。入力データ構造710はToTerm 706からアクセスされ、FromTerm 704へのポインタ、OutMember 722、InMember 732及び関連の型のデータを保持する値バッファ712を含んでいる。

上述したように、本発明の実施例においては、評価は要求駆動であり、タームは、他のタームからそれを要求したときかつその場合に、入力を受け取るのみであることを意味する。説明のための、第1に、ターム706がターム704から値を

要求すると仮定する。この要求に対する適切なメソッドは、ビュー720の定義の一部で、ビュー720からターム706がインスタンス生成される。入力データ構造710は、“From”ターム704のアドレスを含んでいる。これはターム706により使用され、要求された値を供給するタームが識別される。入力データ構造は、要求した値を返すFromTermのOutMember 722へのポインタを含んでいる。（共に、ターム704とメンバー722は、「入力オブジェクト」710を形成し、他のタームからデータを得るに必要な全ての情報を含んでいる。）最後に、FromTerm 704がターム706へ値を提供する方法は、値を要求する一部として、ターム706は、FromTerm 704が値を書き込むために自身のメモリ位置を供給する。FromTerm 704の値は、GetFunc 742の関数を探し、それを、FromTerm 704、OutMember 722及び値バッファ712の引数とともに呼び出すことで、ToTerm 706に対して与えられる。Getfunc 742の関数は入力を、同様の仕方で（この例ではない）評価するように進み、それから適切な値を値バッファ712に格納して、戻る。

コンパイルされたタームに対するGetfunc 742に格納されているゲット関数は、関連するクラスのメンバー関数として実現することができる。その結果、OutMember 722の引数は協議する必要がない。それは、タームへの適切なオフセットのデータ構造がC++コンパイラにより提供するからである。構造的に構成されたタームに対して、OutMember 722は、FromTerm 704の値がFromTermの定義（図示せず）上に格納されるようにする。構造的に構成されたタームへの入力メンバー（図示せず）がある場合、そのタームのために、定義上に格納されているポインタを調べ、特定の入力データにアクセスする。（この例では、タームはFromTerm 704である。）見いだされた入力データ構造は、次に必要ならば評価され、そして、データは入力メンバーにより返される（又は、得られなかったらエラー）。

その上、タームは、FromTerm.Outputs 708として、その値を要求している全てのタームのリストを保持している。すなわち、定義が、供給タームから値を供給できる新しいタームとしてインスタンス生成する度に、供給タームの値への新しいタームの「関心」は、新しいタームの識別をタームのOut_listに追加して、登録する。他の実施例では、自動的に値を全ての関心のあるタームに書き込んでも

よいことを理解すべきである。そしてそれは、タームのOut_listにより、できるようにしている。たとえば、タームの値が更新されたとき、更新された値は、システム中を伝搬することができる。別のものとして、値に対する更新が、適当なときや場合に更新された値を要求する、関心のあるタームに提供する更新の通知を生じる。このように、出力データ構造708は、更新関数中で、実質的な役割を行っている。

入力データ構造710は、他の1つの関数を実行できる。これは、ターム706に対するデータ格納位置として役立つ。この場合、値バッファ712以外、ポインタは使用されない。3つの全ての場合（入力、出力及び格納）、不使用の構成のメモリ空間は、他の目的のために獲得され、これにより、必要なメモリ空間を減少している。

相互に関連したタームのネットワークは評価ネットワークを形成している。タームの入力は、同じ定義の任意のビュー内の任意のタームの出力に、入力と出力との「署名」（“signature”）または型が「適合」（“compatible”）している場合に、（“connect()”関数を介して）関係付けられる。入力に関するデータの型が出力に関するデータの型と同じかそのサブクラスである場合、署名は適合している。関連が定義の間で行われなければならないことを理解することが重要である。むしろ、各定義は、新しい定義のビュー内に（タームとして）インスタンス生成し、関連は、新しい定義のスコープ内でターム間で行われる。

図2に図示されているシステムの一つの追加された特徴は、メリット・メンションである。関連（Connection）と記されている点線のボックス520は、メンバーとタームとの間のいくつかの複雑な関係の仕様を単純化して使用するクラスを表している。関連（connection）クラスは、エンティティではない。それ自身システムの永続的記憶（persistent store）に格納されていない。それは通常、サブルーティンの一時的変数に割当てられており、ターム間の関連の存在を検索したり、関連を繰り返すことに使用する。

システム・アーキテクチャの他の詳細は、図2の定義やアーキテクチャを与えられたこの技術の当業者には自明であろう。さらに、上述した説明は、図2に示

されたアーキテクチャの実装に対する一つの実施例に対するものであることが理解されるであろう。また、図2はそれ自身実施例の一つを示しているだけである。説明の目的のために、しかしながら、図2のエンティティ関係図500の実装の関し、本発明に従った金融手段の構成及び管理に適切な詳細について、記述される。

図5を参照すると、ここでは、本発明の説明のための実施例のグラフィカル・ユーザ・インターフェース50の視覚的特徴とレイアウトを図示している。この実施例のグラフィカル・ユーザ・インターフェース50は、上述のディスプレイ装置40のような表示装置上のワークスペース52内の複数のウィンドウで構成されている。

図5に示されているワークスペース52内に表示されている第1のウィンドウは、ビュー・ウィンドウ54である。この中にターム定義と手段が構成され、表示される。上述のように、手段はタームから構成され、ターム定義リストは複数のタームの定義を含んでおり、ここから手段を構築するためにユーザが選択する。ターム定義リストは、ターム定義リスト・ウィンドウ56に表示される。ターム又は手段を表す多数のアイコンは、ユーザにより先にコンパイルされるか作成されており、手段を構成する過程を簡単化するために提供される。これらのアイコンは、パレット・ウィンドウ57に表示されている。ツールの選択は、構成、手段の視覚面の表示、関係するテキスト、及びよく知られたコンピュータ描画プログラムと同様なグラフィカル・オブジェクトのためにも提供される。これらのツールは、ツール・ウィンドウ58内に表示されている。最後に、ユーザは、構築の間又は後に、様々なエディタを使用して、タームを見、エディットすることができる。様々なエディタとしては、メンバー・エディタ、関連エディタ及び表示エディタがある。エディタの選択は、ターム・エディタ・ウィンドウ60で行われる。

グラフィカル・ユーザ・インターフェース・システムにおいては、ターム、ツール、エディタ等のオブジェクトの「選択」は、通常、ポインタ機器48（マウスのような）を使用して、ポインタ62の位置指定し、ポインタ機器48上にあるボ

タン64を活性化する（クリックする）ことで行われる。その上、メニュー66のよ

うな1以上のプルダウン・メニューが、ディスプレイ上の適切な位置に提供される。別に、ある選択に対するキーボードのショートカット、たとえば“CMD S”（コマンド・キーとsキーとを同時に押下する）でセーブ・コマンドを実行するものが提供される。

手段の作成の例は、適切なシステムの詳細と共に説明される。この最初の例の手段は、スプレッド・オプションとして知られているオプションの特定のタイプである。このタイプを基礎とするオプションは特定の単一の証券ではなく、むしろ、たとえば2つの異なる証券の間の関係の価値である。スプレッド・オプションはヘッジングによく使用される。すなわち、1つの投資が利益を上げるのに失敗した場合、支援のポジションを確保する。これ以上のスプレッド・オプションの詳細な説明は、Mark, Susan Ross, "Derivative Financial Product" pp. 80-81 (HaperBussiness) を参照されたい。

この実施例で構成される特定のオプションは、エイジャックス社 (Ajax Corporation) の株とモノリシック社 (Monolithic Corporation) の株エクイティ・スプレッド・オプションである。説明のために、エイジャックス社は主としてハードウェア事業におり、モノリシック社は、主としてソフトウェア事業にいとす。このスプレッド・オプションは、ハードウェアはソフトウェアより利益になる（又はその逆）という信念にその基礎をおく投資を表現している。オプションはコール・オプションで、それは株を買うオプションを意味している。便宜のために、この例のオプションは、AMESCOオプション (Ajax-Monolithic-Equity-Spread-Call-Option) と呼ぶ。図6は、AMESCOオプションを図示しており、他の図も必要に応じて参照される。この例の手段550の構造は、equity-spread-call-optionの取引定義240であり、エクイティ手段定義556のインスタンスAjax 130とMonolithic 131、エクイティ・スプレッド手段定義150のインスタンスEquity_Spread 190、エクイティ・スプレッド・コール・オプション手段定義200のインスタンスEquity_Spread_Call_Option 230を含む。エクイティ・スプレッド手段定義150は、定義エクイティ・スプレッド・ボラタリティ154のインスタンスEquity_Spread_Volatility 155及び定義エクイティ・スプレッド

NPV 152のインスタンスEquity_Spread_NPV 180が含まれている。エクイティ・スプレッド・コール・オプション手段定義200は、定義Black-Scholes 551のインスタンス224を含んでいる。

Ajax_equity_instrumentターム130は、Monolithic_equity_instrumentターム131は各々、(1) 現在の正味の利益 (Net Present Value:NPV) (2) ボラタリティ及び (3) 手段名を提供する。ボラタリティは、重要な変数に対する手段の一年間の期待した水準の価格の変位の測定である。たとえば、USドルによる現在の日付から1年間のエイジャックス・エクイティ株の価格の期待する水準の変位である。

NPVを定めるために、エクイティ手段は、特定の手段の特定の日付に対する「固定された」価格、又は手段の特定の日付に対する計算された（予測された）価格のどちらかを得る必要がある。「固定された」価格は、NPVが得ようとする日付が、システムがエクイティを実際の引け値にアクセスするための日付である場合である。他の場合は、計算された価格が提供される。図7を参照することで、はっきりと見ることができる。図7は、Ajaxの引け値 (close price) を示している。Ajax株の価格は毎日仕事が終わるときに固定される。Ajax株のテーブル中の過去の株の引け値又は実際の過去のパフォーマンスはテーブル700の領域702に示されている。このため、現在の日付前の日付に対するテーブル中の価格例は、固定?の列中のY (Yes) で示されている通常固定値が返される。これらの固定値は、手動又は、リアルタイム・データ・フィード・サービス704から得ることができる。別に、ユーザは、評価のため、手段のパフォーマンスのモデル化やイベントのシミュレートするために、エクイティの価格を手動で入力することができる。

現在の日付706に対して、及びその日付の仕事が終わる前は、エクイティの値はまだ固定されていない。このように、現在の日付に対して、値は実際の現在値であるかもしれない。データは、リアルタイム・データ・フィード・サービスから返されているかもしれないし、または得たら入力されたかもしれないし、実際の値に不十分にアクセスした場合は、計算されたかもしれない。

システムが実際の歴史的値を有していない日付に対する価格例は、計算された

価格が返される。たとえば、将来のパフォーマンスは、領域708に示されている予測された価格と、領域710に示されている計算された過去の価格に基づいている。実際の価格例が定められる方法は、この開示の範囲外である。しかし、値が計算される方法やアルゴリズムは、価格例定義の一部分である。そのため、値は、他のタームからの入力に基づいて計算することができる。または、値は、予め定め、かつ手動でユーザにより図7に示されているようなテーブルに入力され、または、エクイティ手段556（図6）に入力として直接入力される。いずれの場合も、価格例タームに対する適切な値の入力は、システムの対話的エディタ部分により取り扱われる。

図5に戻り、AMESCOオプションを構築するためのシステムの初期状態を示している。ビュー・ウィンドウ54は、初期にはブランクであり、ターム・リスト、パレット、ツール、及びエディタ・ウィンドウ56, 57, 58及び60が表示されている。

図8を参照して、AMESCOオプションを作成する過程のいくつかのステップを終了後のシステムの状態を示している。オプションを作成する第1のステップは、2つの基礎となるエクイティ手段556と558を作成することである。ビュー・ウィンドウ100は、Ajaxエクイティ手段のために作成される。上述したシンタックスに従って、これには、Demo:Equity_Instrument:Interfaceと名付けられる。タームは、ターム・リスト・ウィンドウ（図示せず）又はパレット（図示せず）から選択される。グラフィック・ディスプレイ・オブジェクト又はこれらのタームを表すアイコンがビュー・ウィンドウ100中に位置付けられる。

図8は、ビュー100に置かれている複数のターム・グラフィックを示している。これらのターム・グラフィックは、Curve_importターム102、ボラタリティターム104及びCurve_rateターム106を表している。また、ビュー100中には、入力メンバーCurve_name108と日付110、出力メンバーInstrument_name 112、ボラタリティ114及びEquity_NPV 116に対するアイコンがある。

ターム・グラフィックのひとつの形態は、四角の領域で、ターム名を表示するテキスト部分、1以上のタームの値を表示する表示部分、スライダー制御、ボタンのようなシステムの値を変える対話部分を備えている。ターム・グラフィック

の作成のような、システムのインターフェースの作成面は、システムの対話的表示エディタ部分で取り扱うことができる。一方、そのインターフェースとユーザの対話は、システムそれ自身により取り扱われる（テキスト編集のような例外を除いて、テキスト編集は、テキスト・エディタに渡される）。本発明のひとつの実施例では、ユーザは、ターム・グラフィックによりどの値が表示されるのかを制御できる機能を提供される。たとえば、タームの個々の値、入力と出力等の表示を選択するために、「スイッチ」はターム・エディタ・ウィンドウ60（図5に示されている）のメンバー部分として提供される。

メンバー及びタームがビュー100中にインスタンス生成され、それらは、エクイティ手段を構成するために、相互に関連付けられている。手段からは、Ajaxエクイティ手段130とMonolithicエクイティ手段131のインスタンス生成される。実施例の一つによれば、ユーザははじめに関連ビュー・オプションをターム・エディタ・ウィンドウ60（図5に示されている）から、たとえば、プルダウン・メニューにより選択する。ユーザは、次に、入力を受けとるターム又はメンバーを、たとえばポインタをターム上に置きマウス・ボタンをクリックすることで、選択する。ターム・エディタ・ウィンドウ60中の関連ビューは、そのターム又はメンバーに対する可能な入力のリストを、ユーザが選択できるように提供する。入力を選択されると、ユーザは、出力を提供するためにターム又はメンバーを選択する。適切なキー又はマウスボタンを押下している間、ユーザは、マウスを動かして、ポインタを出力を提供するターム又はメンバー上に置き、選択された入力に関連付け、その後ボタン又はキーを離す。このようにして、グラフィカル関連118~128がターム又は／及びメンバー間に確立され、表示される。

グラフィカル関連が確立すると、基礎となっている関連が、2つのタームの間でソフトウェア中に確立する。この基礎となる関連は、色々な目的に役立つことができるが、しかし、最も基礎的な形態は、たとえば、ターム間の値の受け渡しのような、あるタームから他のタームへの呼び出しである。このように、視覚的な関連がターム間に確立するばかりでなく、データが、ターム間の基礎となる物理的な相互関連（ソフトウェアを介して）を支持するために、書き込まれる。タームは、たとえば、式内でタームの名前を使用することで、タームの名前を参

照することで関連付けてもよい。このタイプの関連は、名前による関連と呼び、例が下に示される。関連が確立されると、ユーザは、タームまたはメンバーの関連を、上記の過程を繰り返すことで変更することができる。

入力メンバーと適切なタームの間に相互関連が確立すると、タームは、その関数の実行で入力データを使用することができる。たとえば、Curve_importターム102は曲線の名前を入力し、曲線の名前を基にして、曲線を表しているファイルを読み込む。ファイルは、図7の示されているものでよい。Curve_importターム102によりメモリ中に書き込まれたデータは、Curve_Rateターム106により使用できるようにされる。ファイルは、実際のエクイティの引け値を表すデータを備えている。このデータはCurve_Rateターム106で使用され、ファイルに格納されている以外のデータに対する価格の値を計算する。たとえば、補間アルゴリズム（又は他のテクニック）を使用して、実際に得ることができる以前のAjaxエンティティの引け値を定めることができ、また、将来の引け値を予測することができる。NPVは、Curve_rate106の適切なメソッドで、Curve_importターム102とDate member 110により供給されるデータを使用して、定めることができる。

同様に、ボラタリティ（volatility）は、ボラタリティ・タームにより、Curve_importタームにより供給されるデータに基づいて計算される。ボラタリティを計算する方程式は、この分野ではよく知られている。たとえば、Figlewski他、"Financial Option: from Theory to Practice" (Business One Irwin 1990)を参照。この文献は、参照によりここに組み込まれる。別のタームの値を計算するのに、たとえば、ボラタリティ・ターム104のボラタリティの場合は、値は手動により入力されるか他のシステム又はルーティンから導入する必要がある。この場合は、値は人間により入力され、エントリは上記のように入力メンバーにより行われ、または、下記で詳細に説明するようにプライベート・データの方法でおこなわれてもよい。

値をインポートすることに関して、あるフォーマットから他のフォーマットに変換することはよく知られている。インポートするデータは、データを使用できない一つのフォーマットから使用することができるフォーマットに変換する。さらに、本発明に従うと、外部ルーティンを呼び出し、実行することができる。外

部ルーティン（ターム）から返されたータをインポートすることができる。

エクイティ手段556の後に使用するときの簡便のために、ユーザの選択により、たとえば、TIFFファイルや同様のものに格納されているアイコンやグラフィック表示オブジェクト（テキストファイル、スライダ制御など）で構成されているターム・グラフィックを使用してタームをインスタンス生成する。（このグラフィック表示オブジェクトは、図2に示されているグラフィック・エンティティ516の一つの型である。）再び、ターム・エディタ・ウィンドウ60を使用して、今度は、エクイティ手段のインスタンス（ターム）130に対するディスプレイ・エディタ中で、手段の所望のターム・グラフィック表示が選択される。これは、図9のターム・グラフィック130として表示される。ターム・グラフィック130は、ユニークな名前113、たとえば“Ajax”を与えることができる。そして、ターム・グラフィックは、たとえば出力ボラタリティ114、NPV 116、Instrument_name（図示せず）、及び“is a” 112（全てのエンティティ及びタームである）の値を適切に表示する。

AMESCOの例を作成する際に、2つの基礎となるエクイティ手段がある。第1は、上記のAjaxエクイティ手段130である。そして、第2は、Monolithicエクイティ131である。Monolithicエクイティ手段タームは、Ajaxエクイティ手段ターム130と同様な方法で作成され、ターム・グラフィック131（図18に示されている）により表示される。したがって、そのタームを作成する詳細はここでは詳細には述べない。

本例により構成される次の手段は、図10に示されるEquity_spread手段150である。この手段は、それ自身も2つのターム、Equity_spread_NPV定義152のインスタンスであるEquity_spread_NPV 180と、Equity_spread_volatility定義154のインスタンスであるEquity_spread_volatility 155である。Equity_spreadターム150は、入力メンバーEquity1_NPV 156、Equity2_NPV 158、Equity1_volatility 160、Equity2_volatility 162及びCovariance 164を含む。最後に、Equity_spread_instrument 150は、出力メンバーInstrument_name 166、NPV 168、ボラタリティ170を含む。

図11に関して、ここでは、Equity_spread_NPV定義152を詳細に示している。

Equity_spread_NPV定義152は、DoubleMath2lnターム172、入力メンバーEquity1_NPV_in 157とEquity2_NPV_in 159及び出力メンバーNPV_out 174で構成されている。入力メンバーEquity1_NPV_in 157は、図10に示されているEquity1_NPV入力ターム156と関連付けられている。Equity1_NPV入力ターム156は、図8に示されているEquity_Instrument 556のNPV出力ターム116と、AMESCO 204のタームAjax130とのために、関連付けられている。入力メンバーEquity2_NPV_in 159は、図10に示されているEquity2_NPV入力ターム158と関連付けられている。Equity2_NPV入力ターム158は、それ自身対応する、Monolithicターム131のためのEquity_Instrument 556内のNPV出力メンバー166と関連している。DoubleMath2lnターム172は汎用のマス・タームのインスタンスで、「差」の出力をNPV_out出力メンバー174に関連付けている。名前が示唆するように、DoubleMath2lnタームは、2の値A、Bに対して差の関数を実行する。各値は、倍精度浮動小数点数である。出力数NPV_out 174の値を生じる。それもまた、倍精度浮動小数点数であり、図10のNPVターム168に関連付けられている。図11に示されているように、DoubleMath2lnターム172はA、Bの値と引き算の結果を表示する。Equity_spread_NPVターム152は、ターム・グラフィック180により表され、手段の名前とNPV出力メンバー174を図12に示されているように表示する。

Equity_Spread_Instrument150を構成している第2のタームは、Equity_spread_volatility154である。それは、Equity_spread_NPVターム152と同様ではあるが1つの点で異なる方法で、生成される。図13に示されるように、Equity_spread_volatility手段154はフォーミュラ・ターム182、入力メンバーEquity1_volatility_in 161、Equity2_volatility_in 163及びCovariance_in 166、出力メンバーVolatility_out 169から構成されている。Equity1_volatility_in 161は、図10に示されているEquity1_volatilityターム160と関連付けられている。Equity2_volatility_in 163は、図10に示されているEquity2_volatilityターム162と関連付けられている。Covariance_inターム166は、図10に示されているCovarianceターム164と関連付けられている。Volatility_out 169は、ボラタリティ・ターム170と関連付けられている。示されているよ

うに、フォーミュラ・ターム182は、色々な名前、たとえば、Equity1_volatility_in、Equity2_volatility_in、Covariance_inを参照して定義されている。タームをこのように定義することにより、入力メンバーとタームとの間を名前を用いることで関連付けが行われる。上述のグラフィカル関連は行われない。対比のために、グラフィカル関連184はフォーミュラ・ターム182と出力メンバーVolatility_out 169との間で、図13に示される。再び、この手段は、図14に示されているように、アイコン155により表示されてもよい。（メンバー名は、それだけで関連を示唆していない。たとえば、図10のNPV出力ターム168と図16のNPV入力ターム202のような同じ名前を有するメンバーは、必ずしも関連していない。事実、上述のように関連付けられたとき、関連するだけである。）

図10に戻って、2つのタームEquity_spread_NPV 152（ターム・グラフィック180で表示されている）とEquity_spread_volatility 154（ターム・グラフィック155で表示されている）は、入力と出力のメンバーに関連しているように示されている。Equity_Spread_Instrument150は、それ自身、NPV 168やボラタリティ170のような所望の値の表示とともに、ターム・グラフィック190として、図15に示すように表示される。

AMESCOの例として構成するのに必要な最後の手段は、図16に示されているEquity_spread_call_option手段200である。Equity_spread_call_option手段200は、入力メンバーUnderlying_price202、Strike_price 204、ボラタリティ (volatility) 206、Risk_free_rate 208、Value_date 210、及びExiration_date 212、出力メンバーNPV 214、デルタ216、ガンマ218、シータ220及びベガ222で構成されている。これらの出力メンバーは、Black-scholesオプション価格モデル551を実装しているBlack-scholesターム224であるタームの出力に対応している。

上記で説明した方法に対して少しの変化で、Black-scholesターム224により選択した値は、入力に反して、ターム自身の定義の部分とすることができる。たとえば、タームとして定義をインスタンス生成する過程で、ユーザはタームに対して値をエディタ・メンバーを介して入力する機会を提供される。データは、この場合、プライベート・データとして参照される。ユーザにとってそのデータが

自明のものではないからである。プライベート・データは、ターム中に「ロック」される。適切な権限を有するものはデータをアクセスし、修正することが許されている。一方、適切な権限を有しないものは、タームを見ることのみができる。これは、安全のためとデータの統制のために提供されている。

上述と同様な方法で、Equity_spread_call_option手段200は、タームとしてインスタンス生成され、ターム・グラフィック230により、出力メンバーからの選択された値とともに表示される。完成されたAMESCO手段240は基礎となるタームとターム・グラフィック130, 131, 190, 230により表示されている手段とで構成されており、図18に示されている。AMESCO手段240は、それ自身単一のターム・グラフィック（適切な値とともに）で表示することができる。そして、このオプションに対してオプションを表示するような別の手段（図示せず）を構成するのに使用することができる。

上記の過程により構成され、本発明のシステムを使用して、手段は、複数の方法で管理することができる。たとえば、上述のように構成されたAMESCO手段のポートフォリオのような多数の手段を与えて、債務（obligation）、再調整（reset）、支払い、リスク計測や他の手段の属性に関するリストを作成することが望ましい。これは、本発明により、各手段の定義でMake_listタームを構築して、それにより、手段に対する所望のオブジェクト又は属性のリストを作成する。つぎに、ポートフォリオ手段を定義した場合、ユーザはMake_listタームを使用して、所望のオブジェクトまたは属性の「マスター」リストを、各手段のMake_listにより作成されたリストから作成することができる。追加されたタームがマスター・リストを「フィルタ」またはソートし、ある日付以後に期日がかかる負債のような属性のサブセットを作成するために使用される。

本発明の他の特徴は、金融手段の管理と構築のために、データをシステムからエクスポートまたはインポートすることができることである。（データをインポートすることは、簡単に上述している。）統合されたASCIIインポート／エクスポート・メカニズムは、システムのデータベースからデータをASCIIファイルから／へ、インポート／エクスポートすることができる。これにより、既存の金融パッケージでデータを使用することができる。たとえば、利益や損失、資本

要求 (capital requirement)、信用開示 (credit exposure) 等は、前に述べた C*ATS SoftwareからのRisk Manager (商標) により、管理できる。

このメカニズムに特有なものとして、ファイルのフォーマットがある。値を識別するデータとラベルは、同じ行に置かれ (ユーザ特有のデリミタにより隔離されている)、レコードを構成している。

ファイル内で、レコードは論理グループに構成されている。一つのグループ内のデータは、システムのデータベースの単位 (すなわち、キャッシュフローや取引) を構成している。グループは、開始と終了のラベルで識別される。このフォーマットは、BEGIN XXXXとEND XXXXで、XXXXはグループ識別子である。

このフォーマットでインポートとエクスポートを行うシステムのデータベースの例

- +取引 (Transactions)
- +機関投資家 (Institutions)
- +銀行 (Banks)
- +時価 (Quates)
- +市場データ曲線 (Market data curves)

複数の項目 (たとえば、複数の取引) は、各取引ごとに繰り返される適切なグループ識別子とともに、1つのファイルに置かれている。一つの項目 (たとえば1つの取引) 内では、同じレコード (たとえば キャッシュフロー) が繰り返されている。

レコードのリストは、C*AST softwareにより提供されている本発明の1つの実装されたシステムに対するASCIIインポート/エクスポート・ユーザ・マニュアルに印刷されている。この文献は参照によりこの明細書に組み込まれる。グループ内には、キー・フィールドがあり、この実施例では、グループ内に存在しなければならない。

グループ識別子 (及びキー)

Transactions: (取引)**BEGIN TRANBASE****END TRANBASE**

Main part of the transaction (取引の主要部分)

Keys: Code and Leg

BEGIN PL**END PL**

P&L records (can be repeated) (P&L レコード (繰り返すことができる))

Keys: Code, Leg and Date

BEGIN SIDEBASE**END SIDEBASE**

Information about the sides (up to three depending on type of transaction) (サイドに対する情報 (取引の型に依存して3つまで))

Keys: Code, Leg and Side

BEGIN CASHFLOW**END CASHFLOW**

Cashflow records (can be repeated)

(キャッシュフロー・レコード (繰り返すことができる))

Keys: Code, Leg, Side, Coupon Date, Period Date and Type

BEGIN RESETPARM**END RESETPARM**

Reset parameters (リセット・パラメータ)

Keys: Code, Leg, and Side

BEGIN RESETREC

END RESETREC

Reset records (can be repeated)

(リセット・レコード (繰り返すことができる))

Keys: Code, Leg, Side, Reset Date, Effective Date and Period

Date (日付)

BEGIN TRADEBASE

END TRADEBASE

Trade parameters (取引パラメータ)

Keys: Code, Leg, Definition Code and Definition Leg

BEGIN TRADEREC

END TRADEREC

Trade records (can be repeated) (取引レコード (繰り返すことができる))

Keys: Code, Leg and Trade Id

Institutions (機関投資家)

BEGIN INSTITUTION

END INSTITUTION

All information about the institution (機関投資家についての全ての情報)

Key: Party

Banks (銀行)

BEGIN BANKACCOUNT

END BANKACCOUNT

All information about the bank (銀行についての全ての情報)

Key: Bank Id

Quotes (時価)

BEGIN QUOTE
END QUOTE

All information about the quote (時価についての全ての情報)

Keys: Date, Term, Currency, Option, Source and References

Market Data Curve (マーケット・データ曲線)

BEGIN DISCBASE
END DISCBASE

(ディスカウント曲線についての全ての情報)

Keys: Currency and Initial Date.

他の例は、SQLI/Oで、これで、ユーザが、システムからのデータをSQLデータベースに書き込むことができる。これはシステムからSQLステートメントを書くことで達成される。

入力

タームの出力又は文字列データが入力された場合に選択的に付与する128

の入力

出力

SQL I/Oは、各入力を、SQL文字列として評価し、ファイルにプリントし、そのファイルを、Sybaseのようなデータベースに送ることができる。

本発明のある実施例で使用された典型的なパス名は

```

ProgName|INSERT      <table name>
                        (<field1> , <field2> , . . .)
                        VALUES      (value1, value2, . . .)

```

図19は、外部ソースからの外部関数（たとえば、main、実行可能）を呼び出すことの詳細である。たとえば、C、C++、FORTRANのような現代的なプログラミング言語で、シェル・スクリプト・プログラム等として書かれたプログラムは、本発明に従って、システム的能力を強化することができる。図19では、手段300の一部分は、上記に従って作成されている。手段300は外部関数呼び出しターム302を含んでいる。これは、出力304と入力メンバー308を備えている。ターム302は

、パス名のような呼び出す外部関数の識別子312を有している。値は、パラメータとして外部プログラム314に渡される。そして、外部プログラム314からの結果は、外部関数呼び出しターム302に返される。

本発明では、ユーザが、任意の複雑な手段を曖昧性がなく、定義し、価格付けし、評価することができ、手段を他の手段とともにポートフォリオとして管理して、利益と損失、資本要求 (capital requirement)、コスト、信用開示 (credit exposure) を評価し、報告書の要求を取り扱い、リスクを管理し、規則化された要求を合致させ、手段及び取引に関係する全ての文書を安全な環境で作成することを支援することができる。

一般的に、本発明に関する分野の当業者にとって、本発明に対して、構築する際の多数の変更や、大きく異なる実施例や応用例でも、本発明の精神や範囲を逸脱しないものであることが分かるであろう。たとえば、上述において、金融手段を構築し、管理するものに関して説明した。しかしながら、本発明は、金融手段に限定する必要はない。さらに、構築及び管理以外のアプリケーションが本発明により、実行することができる。たとえば、ユーザが手段の構成要素の詳細な知識を要求せずに、手段の動作性能をシミュレートすることができる。(ハードウェア又はソフトウェアの「リバース・エンジニアリング」の概念と同様である。)

このように、ここにおける開示と記述は、説明のためでどのような意味においても制限を意図したものではない。

【図1】

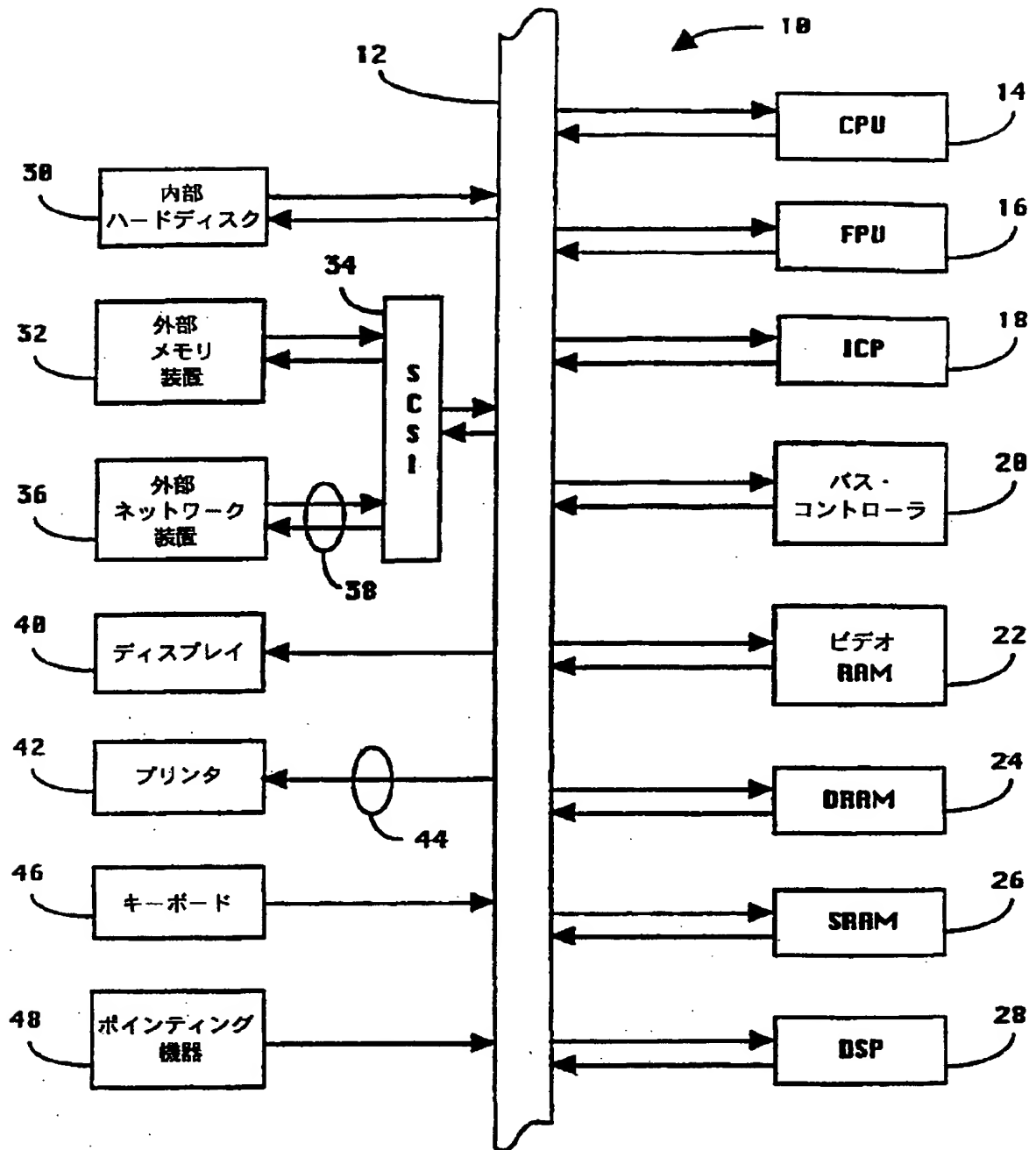


Fig. 1

【図 2】

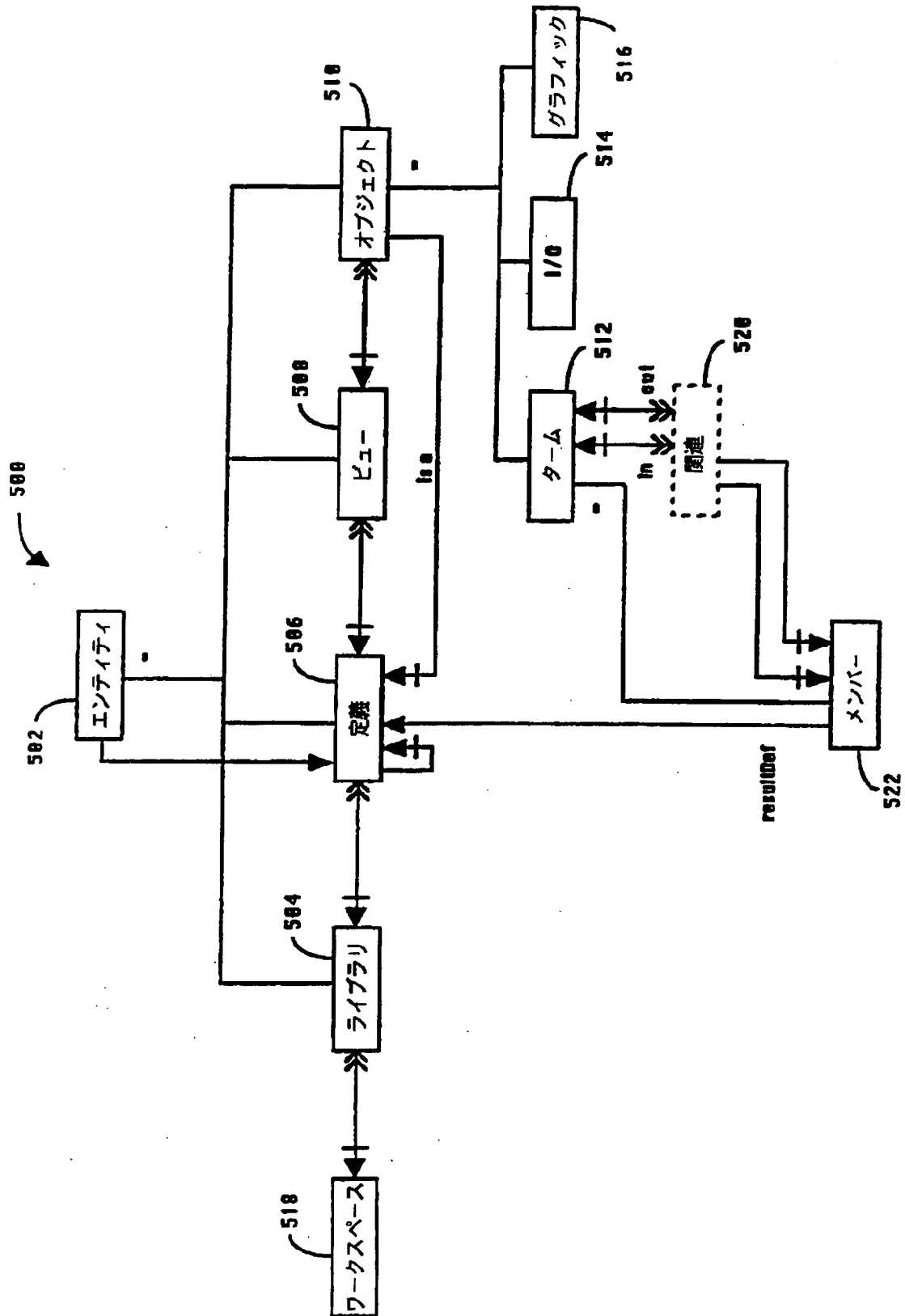


Fig. 2

【図3】

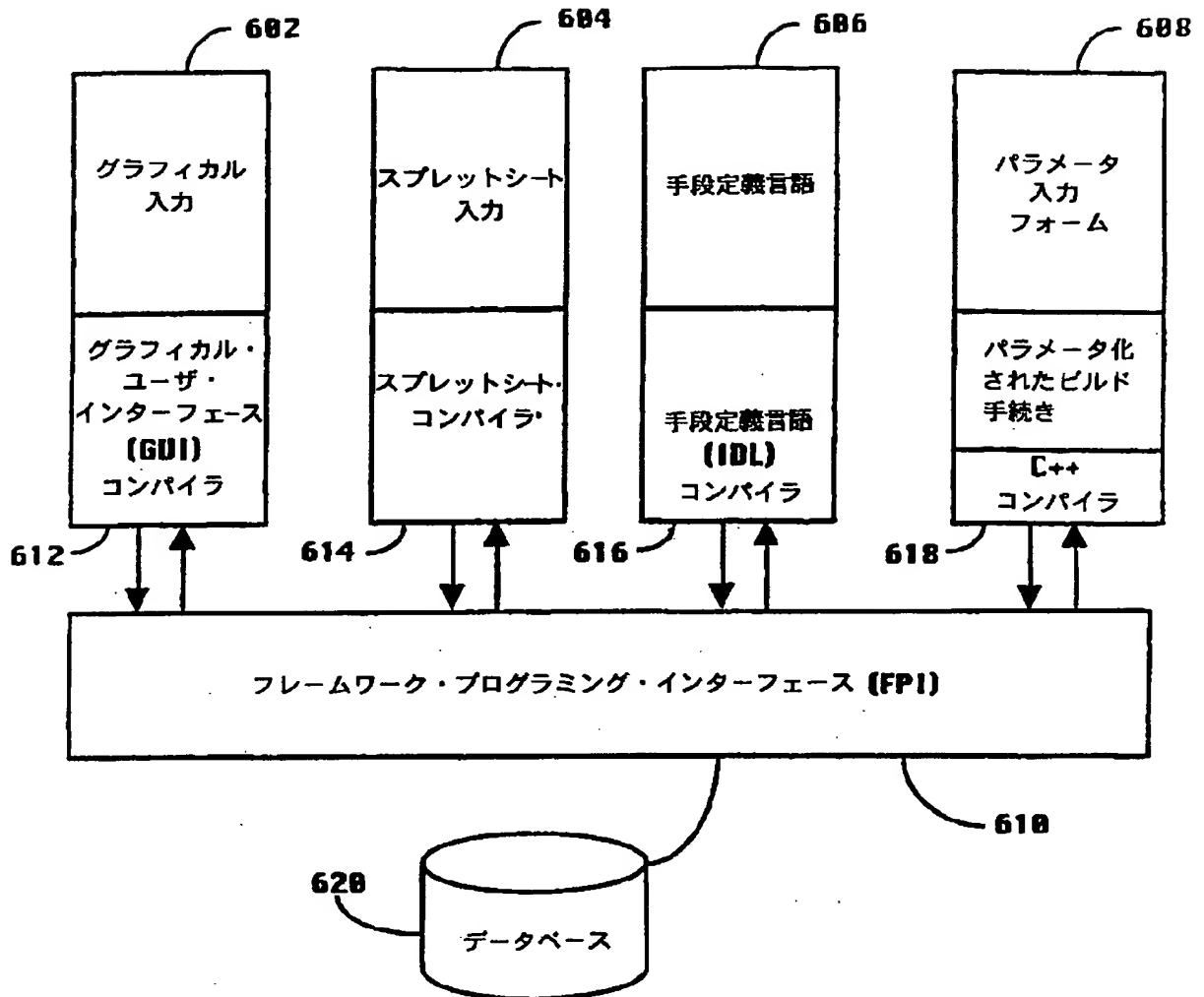


Fig. 3

【図4】

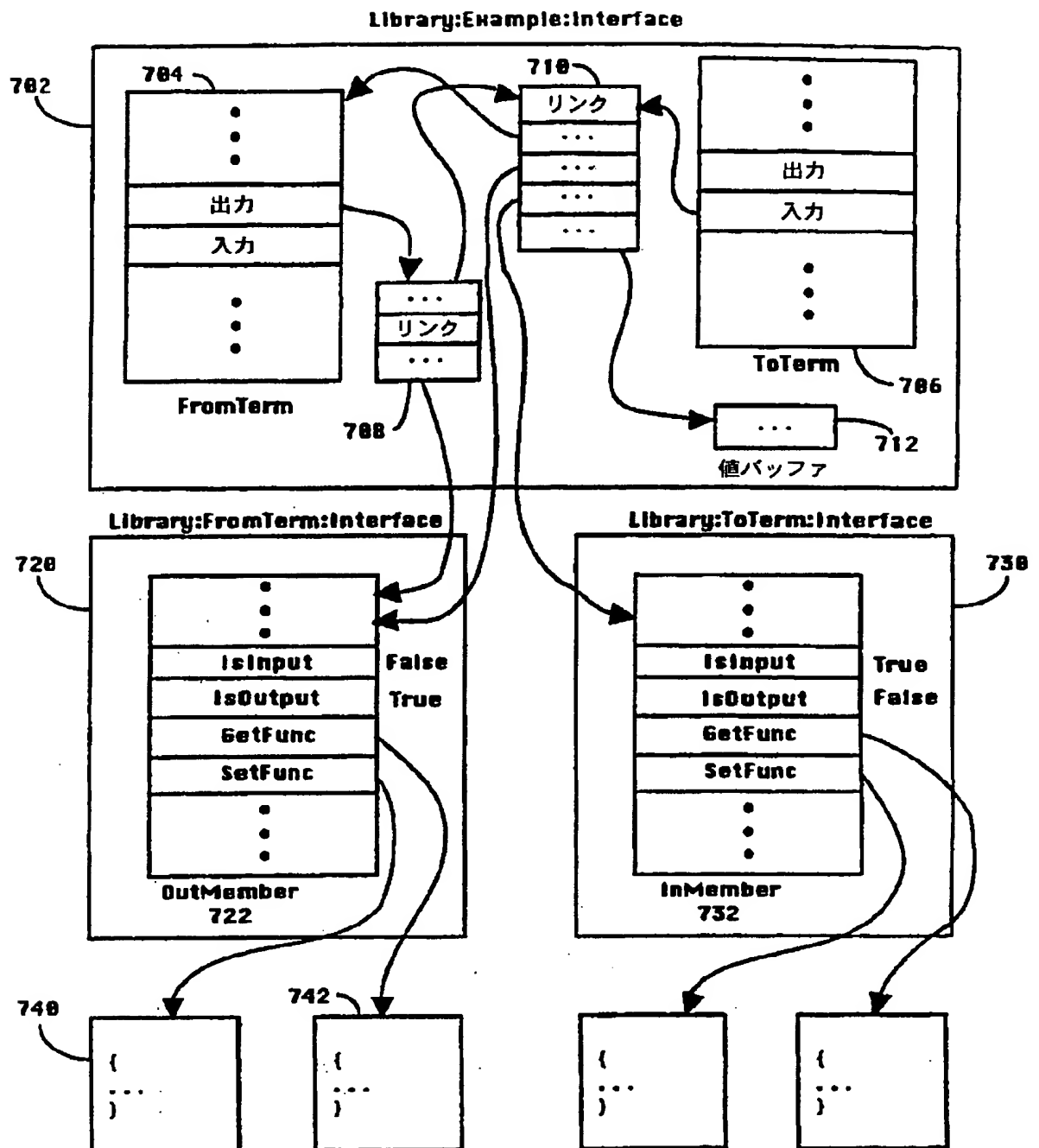


Fig. 4

【図5】

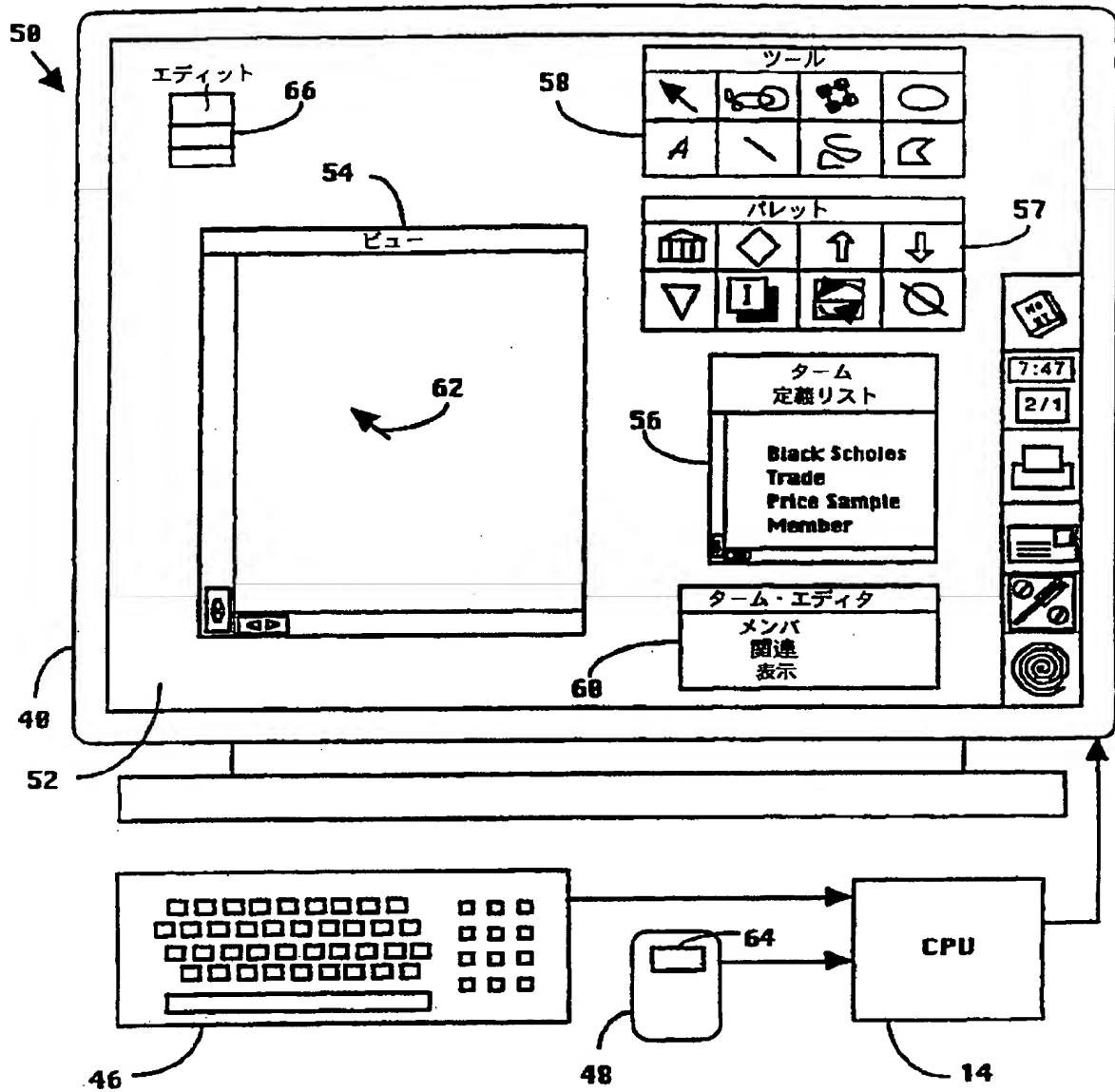


Fig. 5

【図6】

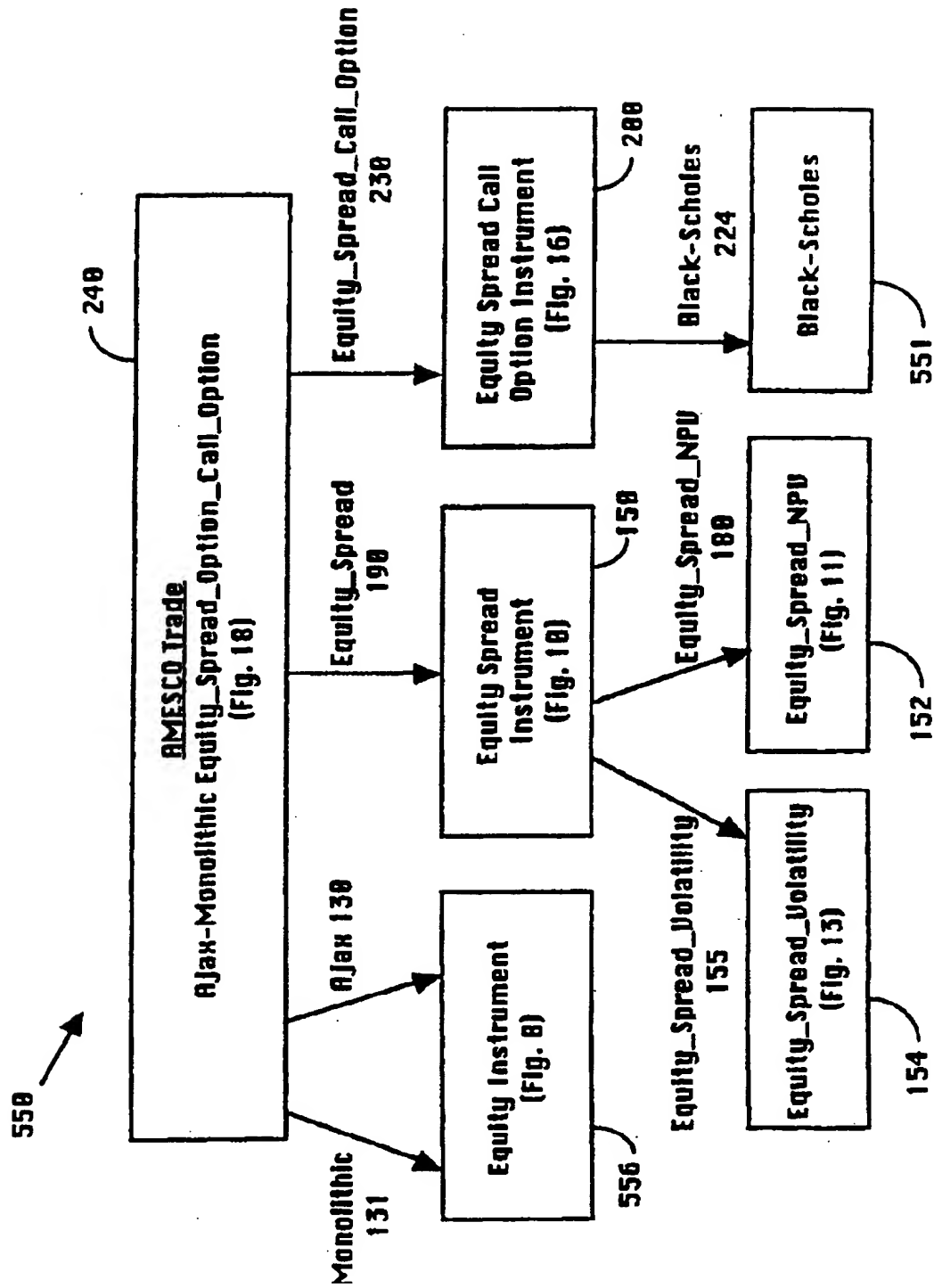


Fig. 6

【図7】

日付	価格	固定?	Flier
•			
•			
•			
1/11	42	y	ACS
1/12	42	y	ACS
1/13	43	y	ACS
1/14	42	y	ACS
1/15	42	y	ACS
1/16	43	y	Service
1/19	43	y	Service
1/20	44	y	Service
1/21	42	n	Service
1/22	44	n	--
1/25	43	n	--
1/26	42	n	--
1/27	44	n	--
1/28	45	n	--
1/29	42	n	--
2/1	43	n	--
2/2	43	n	--
2/3	44	n	--
•			
•			
•			

計算した過去の
パフォーマンス 710

実際の過去の
パフォーマンス 702

現在の日付 706

予測
パフォーマンス 708

リアルタイム・
データ・フィード
704

700

Fig. 7

【図8】

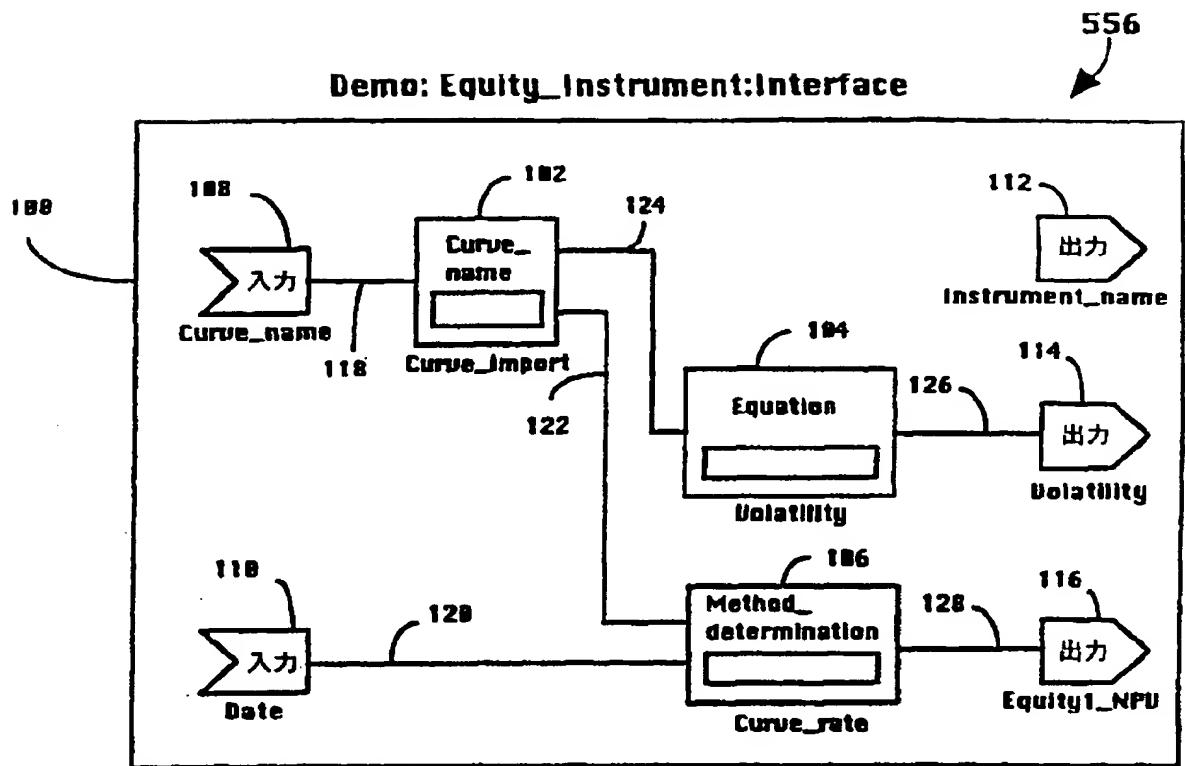


Fig. 8

【図9】

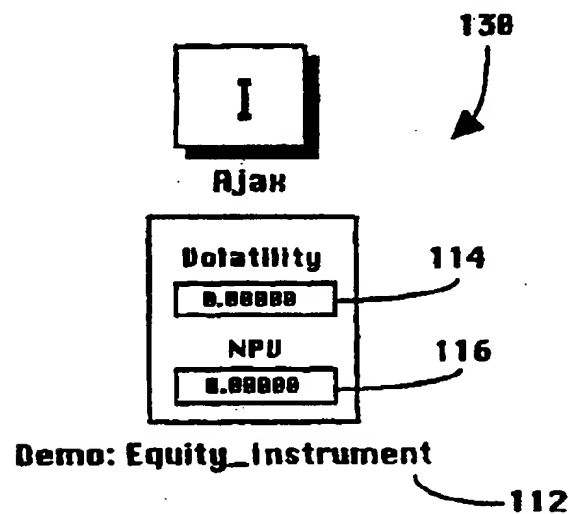


Fig. 9

【図10】

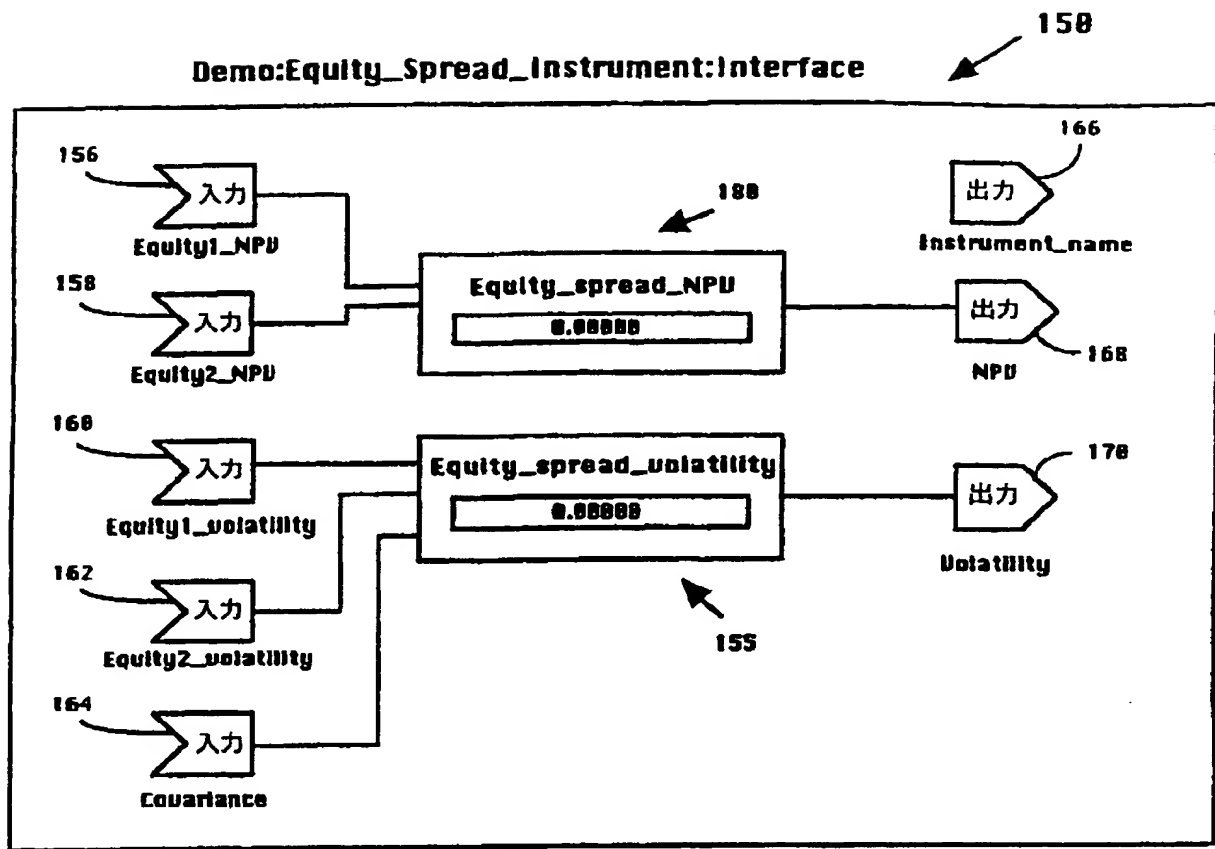


Fig. 10

【図15】

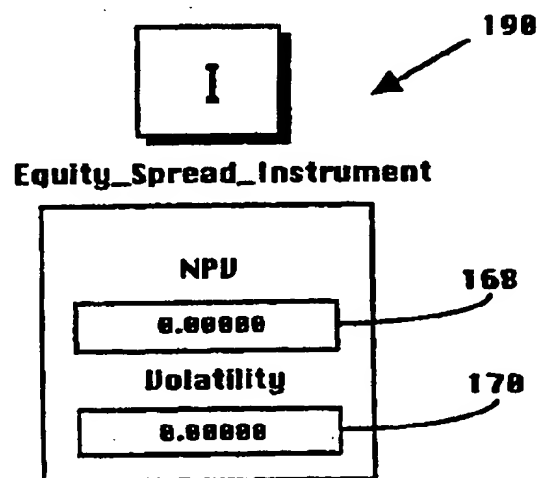


Fig. 15

【図11】

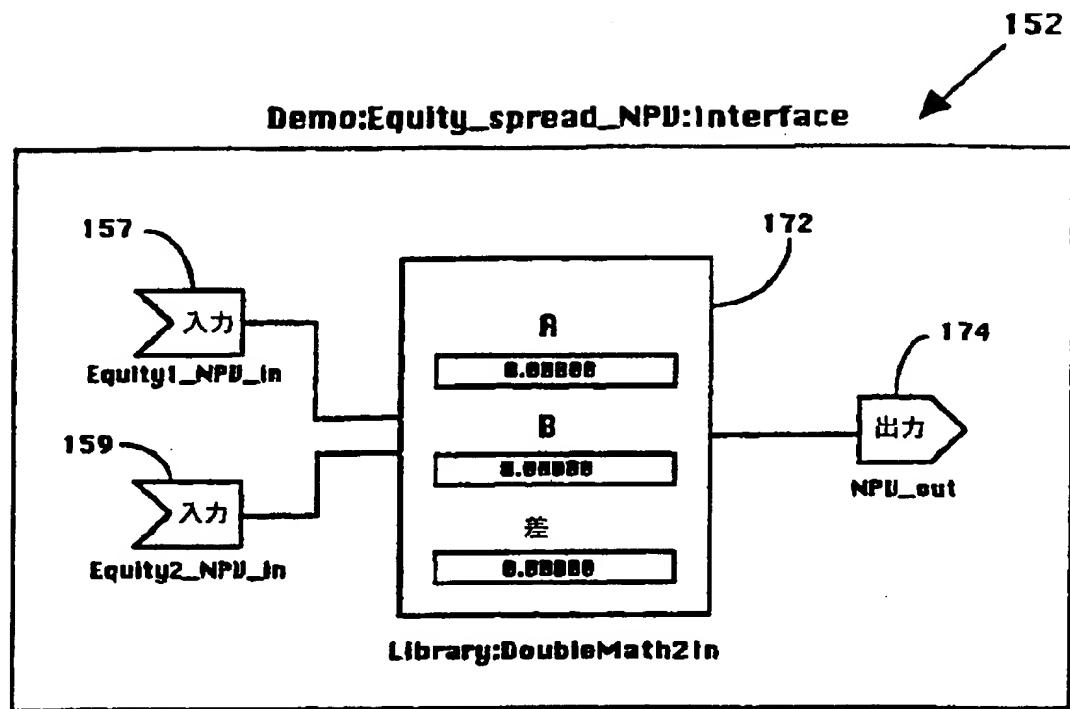


Fig. 11

【図12】

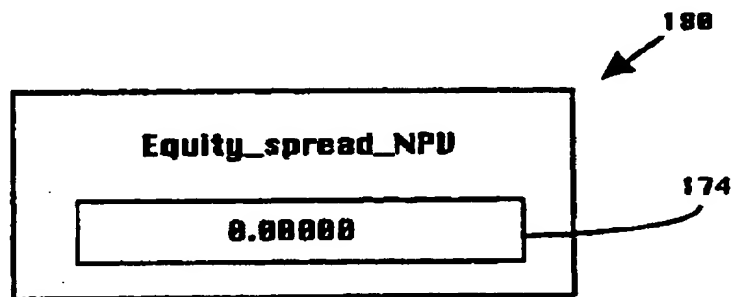


Fig. 12

【図13】

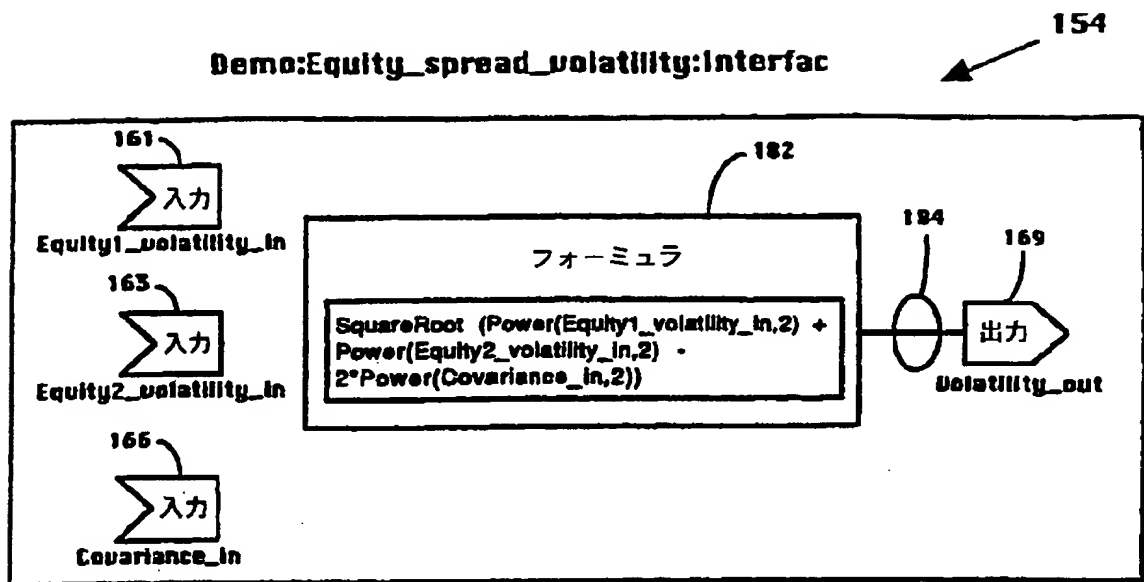


Fig. 13

【図14】

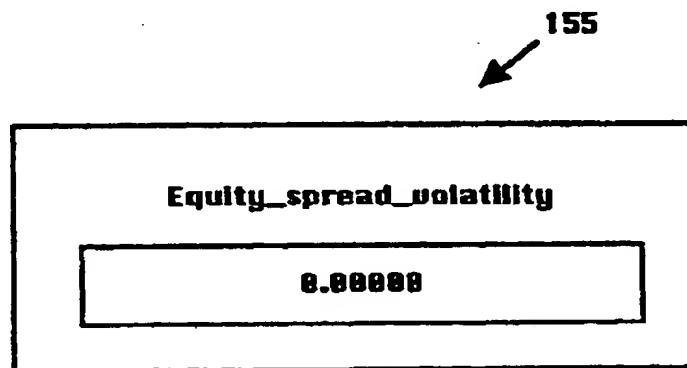


Fig. 14

【図16】

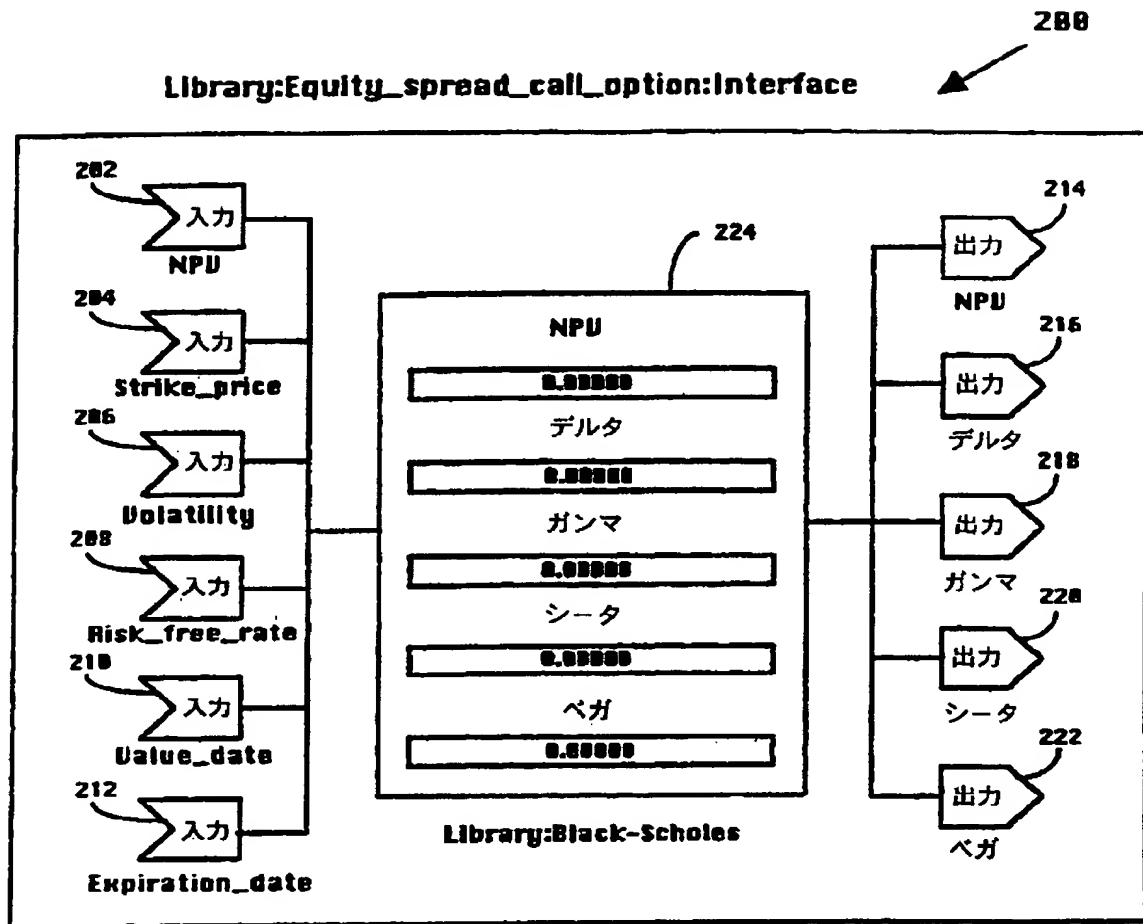


Fig. 16

【図17】

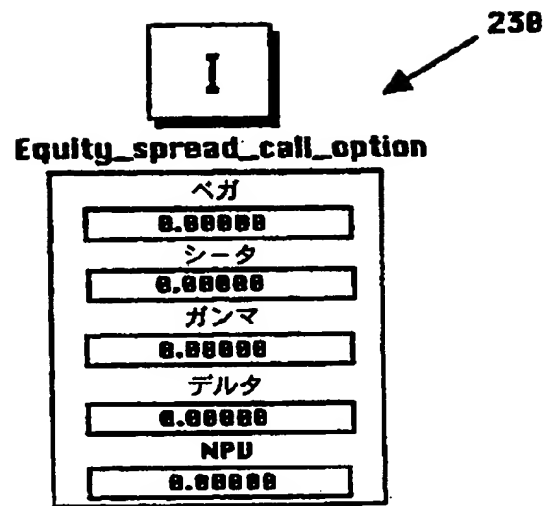


Fig. 17

【図18】

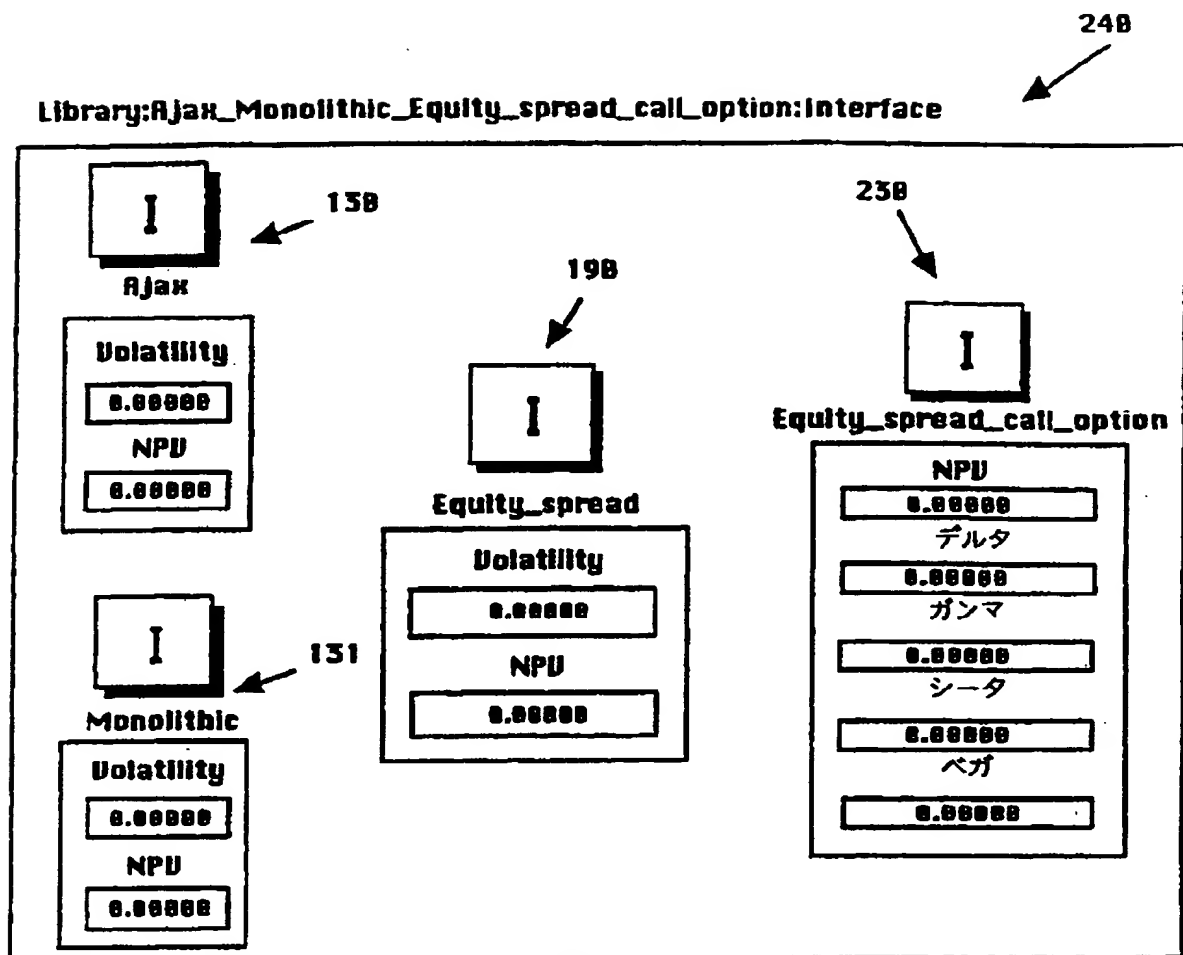


Fig. 18

【図19】

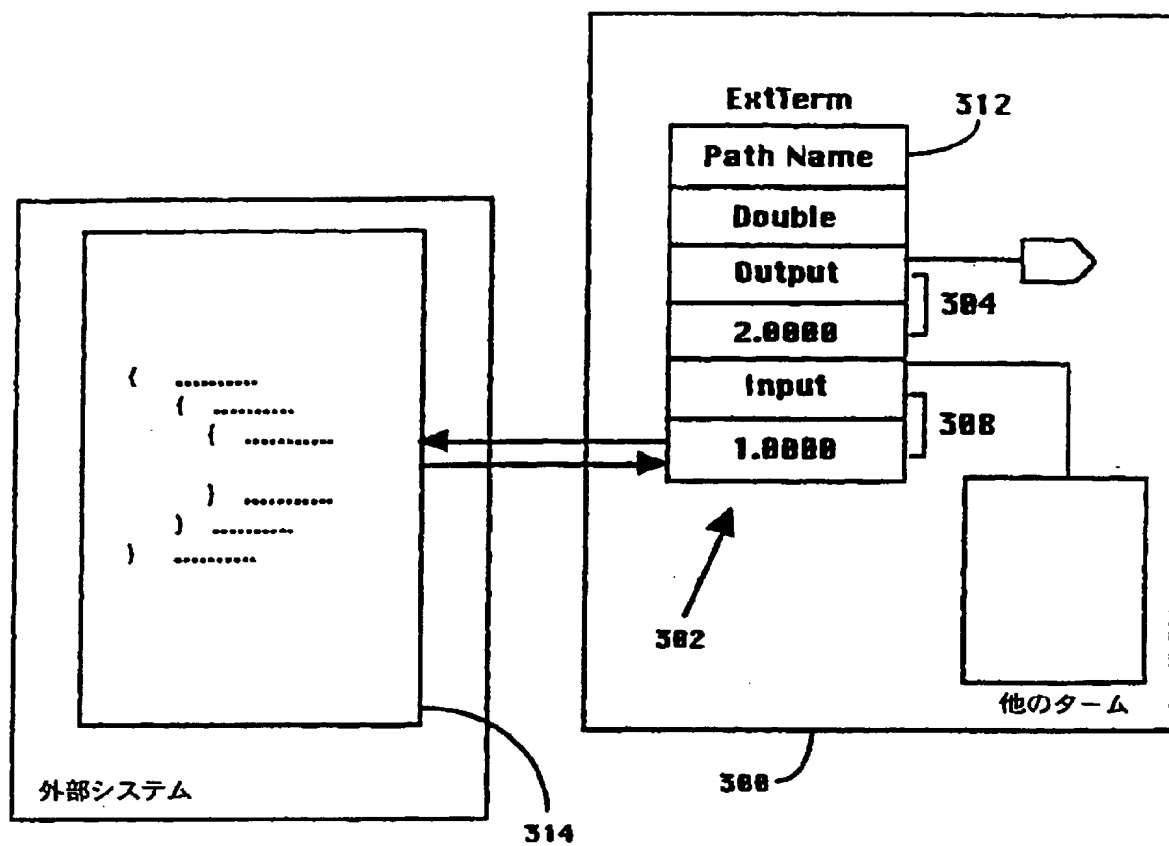


Fig. 19

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No.
PCT/US 94/02468

A. CLASSIFICATION OF SUBJECT MATTER IPC 5 G06F15/30		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 5 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ASSOCIATION FOR COMPUTING MACHINERY - SIGPLAN NOTICES vol. 27, no. 10, October 1992, NEW YORK, US pages 166 - 177 XP000327296 T. EGGENSCHWILER ET AL 'ET++SwapsManager: Using Object Technology in the Financial Engineering Domain' see page 170, right column, paragraph 1 - page 175, left column, paragraph 3 --- -/--	1,2,5-9, 11
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.		
<input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents:		
A document defining the general state of the art which is not considered to be of particular relevance *B* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed		
T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
29 June 1994		05.07.94
Name and mailing address of the ISA European Patent Office, P.O. Box 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 cpo nl, Fax (+31-70) 340-3016		Authorized officer
		Pottiez, M

INTERNATIONAL SEARCH REPORT

Inter. nal Application No
PCT/US 94/02468

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>PROCEEDINGS OF THE 7TH CONFERENCE ON ARTIFICIAL INTELLIGENCE APPLICATIONS February 1991, MIAMI, US pages 168 - 174 XP000298920 M. BENAROCHE ET AL 'An Intelligent Assistant for Financial Hedging' see page 170, left column, paragraph 2 - right column, paragraph 3 see page 172, left column, paragraph 6 - right column, paragraph 2 ---</p>	1
P,X	<p>INFORMATION AND MANAGEMENT vol. 24, no. 5, May 1993, NETHERLANDS pages 267 - 281 A. BANSAL ET AL 'Financial Risk and Financial Risk Management Technology (RMT)' see page 272, last paragraph - page 274, paragraph 1 ---</p>	1-11
A	<p>PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE ON WALL STREET - NEW YORK 9 October 1991, LOS ALAMITOS, US pages 73 - 78 XP000400726 T.M. PATEL ET AL 'The SPLENDORS Real Time Portfolio Management System' see page 73, right column see page 74, right column, paragraph 3 - page 76, left column, paragraph 5 ---</p>	1-11
A	<p>WALL STREET COMPUTER REVIEW vol. 6, no. 6, March 1989, US pages 42, 43, 45, 46, 48, 50, 80, 81 L. KOFLOWITZ 'Hedging Tools provide Portfolio Security Blanket' see the whole document ---</p>	1,7,10, 11
A	<p>EP,A,0 294 187 (CORPORATE CLASS SOFTWARE) 7 December 1988 see abstract -----</p>	

information on patient family members

PCT/US 94/02468

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A-0294187	07-12-88	US-A- 4989141	29-01-91
		AU-A- 1695288	01-12-88
		US-A- 5189608	23-02-93

フロントページの続き

(81)指定国 EP(AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, LV, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, US, UZ, VN

(72)発明者 ベックストローム, ロッド, エイ.
アメリカ合衆国 94301 カリフォルニア
州 パロ アルト エマーソン アヴェニ
ュ 1947

(72)発明者 ガルビン, レイマンド, ピー.
アメリカ合衆国 94087 カリフォルニア
州 サニーベイル ドラ アヴェニ
ュ 755

(72)発明者 オグデン, サンディー, エル.
アメリカ合衆国 94086 カリフォルニア
州 サニーベイル パーム コート 766